**AGENT-BASED MODELING METHODOLOGY FOR ANALYZING WEAPONS SYSTEMS**

THESIS


Casey D. Connors, Major, USA



AFIT-ENS-MS-15-M-132


**DEPARTMENT OF THE AIR FORCE**
**AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

AFIT-ENS-MS-15-M-132

# AGENT-BASED MODELING METHODOLOGY FOR ANALYZING WEAPONS SYSTEMS

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Casey D. Connors, BS, MS

Major, USA

March 2015

AFIT-ENS-MS-15-M-132

**AGENT-BASED MODELING METHODOLOGY FOR ANALYZING WEAPONS SYSTEMS**


Casey D. Connors, BS, MS

Major, USA


Committee Membership:

Dr. J.O. Miller
Chair

LTC Brian Lunday
Member

AFIT-ENS-MS-15-M-132


## Abstract

New weapons system analysis is a field with much interest and study due to the requirement to constantly update and improve the military's capability set. Particularly, as development, testing, fielding and employment of any new weapon system can be quite costly, justifications of acquisition decisions need to be made carefully in order to provide the capabilities needed at the least possible cost. Getting as much information as possible to make these decisions, through analysis of the weapons systems benefits and costs, yields better decisions. This study has twin goals. The first is to demonstrate a sound methodology to yield the most information about benefits of a particular weapon system. Second, we wish to provide some baseline analysis of the benefits of a new type of missile, the Small Advanced Capability Missile (SACM) concept, in an unclassified general sense that will help improve further, more detailed, classified investigations into the benefits of this missile. In a simplified, unclassified scenario, we show that the SACM provides several advantages and we demonstrate a basis for further investigation into the tactics used in conjunction with the SACM. Furthermore, we discuss how each of the chosen factors influences the air combat scenario. Ultimately, we establish the usefulness of a designed experimental approach to analysis of agent-based simulation models, which yields a great amount of information about the complex interactions of different actors on the battlefield.

*This work is dedicated to my wife, who is the most supportive, caring, and hard working person I know.*

# Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. J.O. Miller, and my reader, LTC Brian Lunday, for their guidance and support throughout the course of this thesis effort. I am grateful to have had their insight, experience, and encouragement throughout this process. I would also like to thank my sponsor, Robert Cisneros, from Lockheed Martin, and all of the folks at the Air Force Research Laboratory, in particular Dave Panson, Brian Birkmire, Chris Linhardt, Matthew Smitley, and Adrian Mazzarella, for the support, feedback, data and programming assistance. Finally, I would like to thank the faculty at AFIT for challenging me, providing guidance, and teaching me to be a better Operations Research Analyst.

Casey D. Connors

## Table of Contents

Page

# List of Figures

Page

Page

# List of Tables

# AGENT-BASED MODELING METHODOLOGY FOR ANALYZING WEAPONS SYSTEMS

## I. Introduction

### Background

Introduction of a new missile into the complex system of air combat necessarily causes major changes to the outcomes of air combat. The Air Force wages air combat to achieve certain strategic objectives using specific air combat tactics. The objectives can be to gain air superiority in theater, or destroy strategic enemy ground targets, etc. (Bullock, McIntyre, & Hill, 2000). An emerging strategic objective is to attack the boost phase of ballistic missiles using the Airborne Weapons Layer concept (AWL) (Corbett, 2013) and (Rood, Chilton, Campbell, & Jenkins, 2013). These requirements have led to recent missile technologies that are agile enough to perform multiple traditional and emerging roles. This paper explores a potential methodology for analyzing the missile system in a constructive agent-based simulation model.

The main characteristics of the new missile technology examined in our research include hit-to-kill technology in which the missile uses a kinetic warhead to attack the target, agility in that the missile's guidance, propulsion, and control surfaces allow it to maneuver more flexibly towards a target, and a smaller size allowing each fighter to carry more missiles. These new weapons have the potential for dramatically changing the range of possible tactics and mission roles allowed.

Complex systems are typically modeled through simulation in order to provide comprehensive information about how the system performs. Agent-based modeling has

the potential to provide additional information about the potential use of a weapon due to the inherent learning or adaptive characteristics of the agents in the simulation model (Bullock, McIntyre, & Hill, 2000).

## Problem Statement

In order to better define the benefits provided by a missile with more flexible capabilities, an analysis methodology is required to show the effects to the overall air combat system of the factors of improved agility, decreased size, and hit to kill capability. The main problem addressed in this thesis is identifying an appropriate methodology for studying a new weapon system, specifically in this case a missile system. Our research seeks to show an analysis method of the effects of a new weapon on tactics and combat decision making by modeling flexible agent behaviors in a mission level combat simulation.

## Research Objective and Scope

### Objective.

The objective of this study is to develop a methodology to analyze a new type of missile system and explore the range of tactics to employ this missile. Specifically, the objective is to quantify the significance and contribution of particular characteristics of a new missile system over existing missile systems using a statistical and practical comparison approach. This methodology also applies in a more general sense to new platform delivered weapon systems and perhaps even new types of sensors and communications systems. Below, we describe the overall system in terms of the

components, such as platform, weapon, sensor, etc. Following that is a sketch of the agent-based simulation approach.

**System Description.**

Air combat is a complex system where there are opposing forces with opposing objectives (Bullock, McIntyre, & Hill, 2000). Air combat is conducted by opposing forces using attack aircraft, defense aircraft, and defense ground platforms using a variety of systems, from guns and energy weapons to missiles. The focus of this study is the missile weapon system as a component of the air combat framework. For the purposes of this study, search and acquisition sensors for all fighters are held to be generic targeting sensors. Fighter platforms are held as fourth generation fighter aircraft.

The focus of the investigation is on weapon factors such as range, speed, turning radius, weight, air-to-air capability, air-to-ground capability, or both, end-game guidance precision, accuracy, and type of warhead. Additionally, we use pilot and commander behavioral modeling to study different air combat tactics in relation to these weapon factors.

**Approach.**

We choose a simulation model to study the complexities of air combat. Specifically, we use agent-based simulation because of the ability of simulated agents to model a complex adaptive system (Bullock, McIntyre, & Hill, 2000).

There are several statistical methods available for conducting analysis of simulation models (Law, 2007). The main statistical approach taken to analyze the simulation output data is a designed experiment in order to more fully understand the significance of the factors involved in the system.

## Investigative Questions/Issues

### Issues and Essential Elements of the Analysis.

All studies begin with a breakdown into the main investigative questions that define the problem under study. For this study, the focus is on the missile system component of air combat. The following questions define the exploratory space:

1. **What is the benefit of being able to carry more missiles? How does size/weight of missile affect mission outcomes?**

New missile technologies are consistently providing more and more compact electronics guidance packages, control mechanisms, and warhead capabilities. Combined, these new missiles are smaller and lighter. This is not without tradeoffs, such as speed, and the need for increased accuracy and precision within the on-board guidance systems.

2. **What is the proper mix of weapons? How does mission mode (air-to-air, air-to-ground) affect mission outcomes? Is there a benefit to carrying a mix of weapons?**

Having multi-role missile systems means being able to strike a wide range of target types, from ground to air, fast moving to slow moving. However, tradeoffs in the missiles systems to include newer technologies such as those discussed above in speed, etc., suggests that a strict replacement of traditional single-role missiles may have a detrimental effect. In other words, there may still be a need for the faster medium range air-to-air missiles.

3. **What new tactics are possible given new weapon characteristics? Do tactics change over the range of each of the characteristics of the new missile type?**

With these new missiles capabilities, fighter pilots may no longer need to conduct some of the aerial maneuvers required with current missile capabilities or there may be more optimal maneuvers when engaging enemy fighters or ground targets. This can effect pilot training and fighter doctrine extensively.

**Indicator Measures of Effectiveness and Performance.**

To explore the different issues, we develop several response measures. The basic procedure is to create different scenarios with each factor of the system set at different levels and then execute multiple replications of the simulation model for each scenario, measuring specific responses. These responses correspond to measures of effectiveness and performance (MOE/MOP) of the system. These measures are indicators that answer the questions from section 1.4.1.

**MOE 1: Time to service Target set for a given mission.**

This is the average time until the Blue force completes clearance of the sweep area. We do not specify that the Blue must destroy every target within the sweep area, but rather this is the time until all Blue forces arrive at the end of the designated sweep route after having engaged/destroyed as many of the opposing forces as possible. The time it takes to complete the mission is an indicator that provides information about questions 1, 2, and 3 in section 1.4.1. This MOE shows some effect in terms of efficiency gained or lost when using the new missile technology.

**MOE 2: Percentage of target set destroyed at scenario end.**

This MOE is a measure of overall mission effectiveness. The goal is to ensure the mission area is as clear of opposing forces as possible. We calculate MOE 2 by finding the number of opposing force targets destroyed during the mission, then dividing by the total number of targets at the scenario start. This MOE provides information about questions 1, 2, and 3 in section 1.4.1 and addresses effectiveness gained or lost.

**MOE 3: Weapon effectiveness.**

In comparison with different levels of factors settings, it shows the improvement or decline in effectiveness and efficiency of the missile system, though we term it "effectiveness" for brevity. This MOE is the number of weapons required to produce one enemy kill. Weapon effectiveness can be calculated using the average number of weapons fired along with numbers of targets destroyed. This MOE provides information about questions 1, 2, and 3 in section 1.4.1.

**MOE 4: Standoff of Engagements.**

This MOE is a measure of the average distance that Blue agents deploy weapons against targets in each scenario. The set of standoff performance measures, such as average engagement distance, may be substantially different from the baseline scenario due to the interaction of the range of a missile and the size of the missile. Smaller missiles cannot carry as much fuel, and therefore usually lack the range or the speed of larger missiles. This MOE is an indicator of the type of tactics employed by the agents in the model and addresses question 3 from section 1.4.1.

**MOE 5: Blue side Vulnerability.**

Blue side vulnerability measures the number of hits the opposing force successfully makes on Blue agents. In order to observe this response, we set the Blue agents to invulnerable. This allows us to see how many times the agents place themselves into risk situations based on the weapon loads they are carrying, the tactics they are using and the composition of the Red force they are facing. The MOE provides additional information on question 1, 2, and 3 from section 1.1.

**MOE 6: Qualitative Engagement Results.**

This MOE is a more subjective measure used to capture insights learned from viewing the playback of numerous design point scenarios, including those scenarios whose response appear to be outliers as well as scenarios whose treatment combinations are more in the middle of the design region of the simulation experiment. While we use the playbacks more for verification of the simulation model and troubleshooting potential errors, the results do have some impact on our view of the validity of the model and interpretation of the statistical analysis.

## Constraints, Limitations, Assumptions

**Constraints.**

The first constraints imposed are that the model used to conduct this research is a mission-level scenario as opposed to a higher-level theater wide or strategic scenario and that the scenario is of a limited time duration. This constraint follows directly from the availability of agent-based combat models and time limitations detailed below. Combat model development is a lengthy process and long, detailed scenarios in higher-level

agent-based constructive simulations are processing intensive. To provide an acceptable scope for constructing and analyzing a model sufficient to gain insight into a weapon system and demonstrate a methodology for analyzing a new missile technology in the available time, we chose a mission-level model of a limited simulation mission time for this research.

Additionally, the models created are constrained to one mission type. Again, the time available to conduct this research necessitates that combat model development be simplified. The complexity of the analysis of each factor's significance in contribution to multiple responses of interest increases immensely if multiple mission types are studied.

One final constraint is that this research is limited to unclassified information. Therefore, we use less detailed data, in terms of missile capabilities and air combat tactics, of all the systems involved. Classified research can be conducted using these methodologies, but is again beyond the time available for this study.

**Limitations.**

The main limitation for this study is the time available to complete this research.

**Assumptions.**

The first assumption is that interactions between platform sensor and weapon performance are negligible. In other words, each scenario maintains a constant generic platform sensor. This assumption refers to the sensor on the aircraft, not guidance package available on the missile. An addendum to this assumption is the assumption that Blue will always have superior sensors, such AWACS, and command and control networks. Given this assumption, the model implements a slight advantage in range of

the fire control sensor to the Blue side, but does not implement a complicated command and control network or utilize AWACS for an integrated air picture.

Another assumption is that each flight, both Blue and Red come in pairs (number of aircraft is multiple of two.) Additionally, red ground units are in groups of four under one command and control element and all threat ground units in the scenario are SAMs, Command and Control, or SAM radars.

A final assumption is that it is sufficient to demonstrate this methodology for analyzing a new weapon system on a smaller, less complicated scenario. Specifically for this study, we use a “sweep”-type mission in which a group of Blue aircraft moves through battlespace with the mission of clearing the zone of enemy aircraft and air defense systems.

**Thesis Overview**

Chapter 2 is divided into three sections that include Department of Defense (DOD) use and classification of models and simulation, agent-based modeling, an overview of the Analytic Framework for Simulation, Integration, and Modeling (AFSIM), and some statistical analysis methods used for simulation analysis including some basic experimental design information. Chapter 3 discusses the methodology, scenario, and analysis techniques used in this research. Chapter 4 provides analysis of the simulation model to illustrate the methodology in Chapter 3 and to provide some level of verification and validation. Finally, Chapter 5 provides the main conclusion and recommendations regarding this analysis methodology using an agent-based simulation model to analyze a combat system.

# II. Literature Review

## Overview

This research is an effort to define a methodology for use of agent-based modeling (ABM) to analyze the effectiveness of a new type of missile in air combat. We review Department of Defense (DOD) Modeling and Simulation modeling classifications, agent based modeling and complex adaptive systems, including different agent behavioral architectures, and various statistically based methodologies for analyzing simulation output. This review includes a survey of several past works using a simulation to model combat with a focus on analyzing a particular weapon system. The focus throughout is on providing a scientifically sound method for discovering the differences between a combat scenario with current technologies and the scenario with addition of the new weapon system.

## Department of Defense (DOD) Models and Simulation

Generally, simulation models are classified by whether they are dynamic or static (simulation includes the component of time or not), continuous or discrete, deterministic or stochastic (simulation does not include random effects or does), and descriptive or prescriptive (simulation either describes the system or is intended to provide a set of optimal settings for the system) (Hill & McIntyre, 2001) (Law, 2007). We are most interested in the set of simulation models that are dynamic, stochastic, and descriptive in nature for the particular problem of analyzing a new weapon system. Prescriptive models can also be useful for discovering the best tactics or weapon system characteristics, such as how many of each type of weapon in a weapon mix problem, that maximize the

effectiveness of the Blue force in a given scenario. A number of useful simulation models are classified as discrete event simulations in that they perform calculations of the system state at discrete points in time based on scheduled events (Hill & McIntyre, 2001).

DOD classifies simulation models according to the way the model is used and the model level of resolution. Simulation models are classified into three broad categories within DOD (Hill & McIntyre, 2001). Live simulations are training exercises with troops and equipment conducting missions in a real environment simulated to look and feel like real combat situations, such as the National Training Center (NTC) at Fort Irwin, California. Virtual simulation models entail troops/pilots working in simulators, such the Close Combat Trainer for ground troops, various M1 SEP, M2 Bradley and HMMVW vehicle simulators, and various aircraft simulators. These simulators are built to mimic as closely as possible the operation of the real vehicles. Finally, constructive simulations are closed models run without any human interaction. There are also hybrids of virtual, live, and constructive simulations, in that an experiment or training scenario is run as a confederation of these three types of models. The constructive class of models is the class applicable to this research.

Figure 1 is a diagram of the DOD model hierarchy. The diagram shows the classification of each simulation model according to its level of aggregation and resolution. Aggregation is defined by the DOD as “the process of grouping entities while preserving the salient effects of entity behavior and interaction while grouped” (DOD Models and Simulation Coordination Office, 2014). Resolution is defined by DOD as “the degree of detail used to represent aspects of the real world or a specified standard or referent by a model or simulation” (DOD Models and Simulation Coordination Office,

2014). Aggregation and resolution are inversely proportional to each other, as the level of aggregation goes up, i.e. entities are consolidated from individual instances into higher level units, the amount of resolution of the model goes down, i.e. less detail per individual base level entity, and vice versa. An example of this is several infantry Soldiers modeled as conducting combat operations is of higher resolution than the model of an infantry company, made up of Soldiers, that does not explicitly calculate the actions of the individual Soldiers in the company.

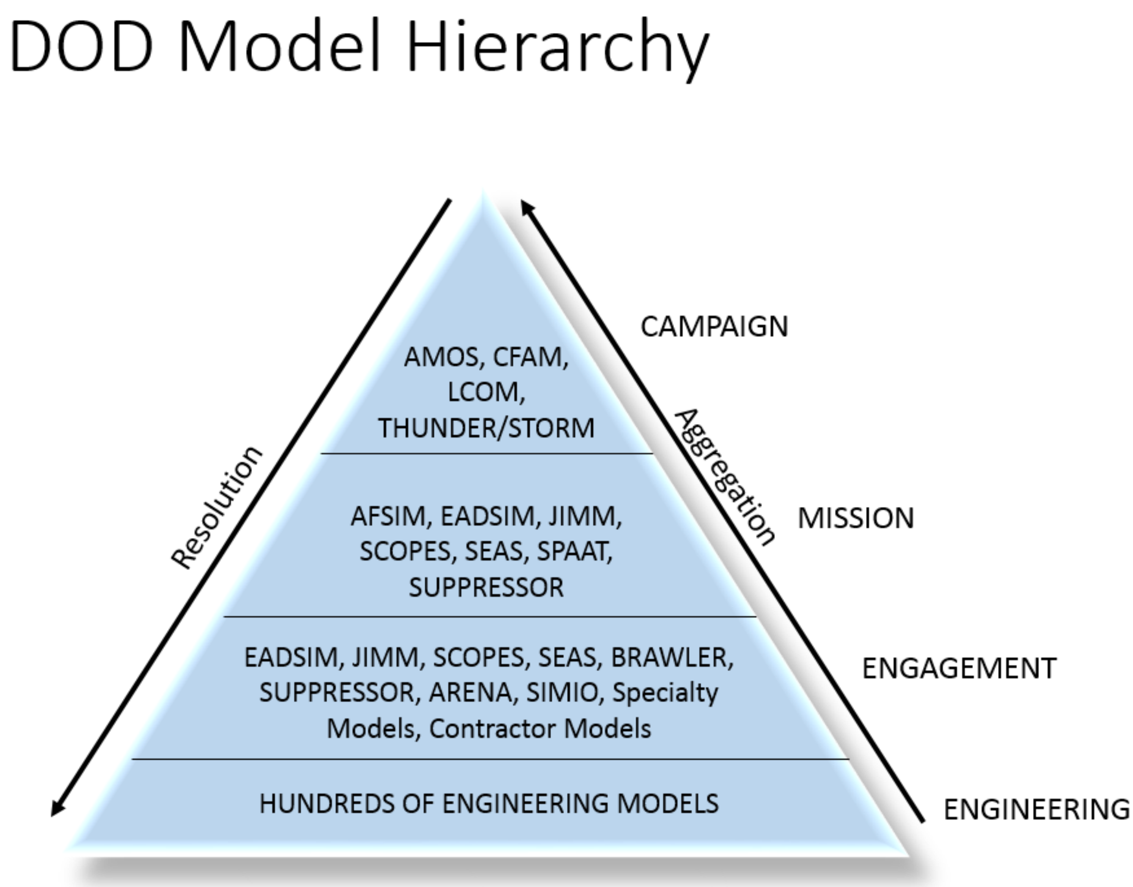



**Figure 1: Department of Defense Model Hierarchy with Several Exemplar Models for each level (Adapted from (Hill & McIntyre, 2001))**

Campaign-level models are the highest aggregation, providing a simulation of only aggregated units at a top level.  This type of simulation is useful for providing information on the actions of large units within a longer period (over several weeks or months).  The campaign level model rolls up the results of many missions and operations using less detailed calculations.  Because of the size of the units involved and length of time, aggregation means that the results are not as detailed, but the simulation model's computation time is drastically reduced.

Mission-level models provide a more detailed simulation of entities over the course of single missions.  These models are less aggregated and therefore more computationally intensive, meaning that the simulation model takes longer to run.  These models usually simulate a system over hours of time rather than days and have much higher resolution, tracking individual platforms (entities) and providing feedback on entity actions and state.

Engagement-level models are of the highest resolution and are used to model specific engagements.  The scenario time lengths for these models is usually minutes and involve a single set of circumstances, such as one exchange of missiles for two opposing sets of fighters.  Finally, engineering-level models, of which there are many, are usually constructed by engineers to help understand the dynamics of a particular entity, such as a missile in flight.

Examples of each of the types of models in the hierarchy are shown in the diagram at Figure 1.  These examples are primarily Air Force models, but there are many more simulation models across the DOD.  The focus of this study are mission level

models, which provide a fair amount of fidelity, while allowing tractable computation times for a short duration study.

Organizations across the DOD use simulation and modeling for training, acquisition of new equipment and weapons, and research into new tactics, techniques, and procedures. The purpose of our research is to study the use of a new missile system within an air combat environment. Therefore, the focus is on developing a mission level scenario in a dynamic, stochastic, discrete event simulation model. For this study, we use the Analytic Framework for Simulation, Integration, and Modeling (AFSIM) to develop out model. We discuss AFSIM in more detail in section 2.3.6.

## Agent-Based Simulations

### Complex Adaptive Systems and Agent-Based Modeling.

According to the DoD Models and Simulation Coordination Office's M&S Glossary, adaptive systems are those "able to modify its behavior according to changes in its external environment or internal components of the system itself" (DOD Models and Simulation Coordination Office, 2014). Middleton defines complex adaptive systems as "dynamically interacting open systems, characterized by "emergence", with non-linear and chaotic behaviors" (Middleton, 2010). Complex adaptive systems are ones in which individual decision-making entities act and react to the environment around them as they attempt to accomplish their specific goals. In agent-based modeling (ABM), the decision-making entities, such as pilots or Integrated Air Defense System (IADS) crew, which we call agents throughout the rest of the paper, "learn" from the environment around them by reacting according to adaptive rules or models. This can lead to the

emergence of complex combinations of behaviors. An important point to note here is that ABM is not mutually exclusive of a discrete-event simulation. ABM can incorporate continuous or discrete-event timing. Most ABMs use discrete-event timing, as we do in our simulation model.

ABM is a form of simulation modeling that uses artificial intelligence techniques to provide agents in a simulation with goals and rules for how they may act towards attaining those goals. Parunak, Savit, and Riolo (1998) published similarities between ABM and the more traditional equation based methods of simulation and developed criteria for choosing one method over the other for a particular problem. The three main criteria that Parunak, Savit and Riolo present concerning the choice of modeling technique involve model structure, system representation and the degree of validity, coupled with the simplicity, of the overall model. ABM is best suited for a model structure where the basic state of the system depends on the behaviors of individual agents within the system. Because air combat is a complex adaptive system in which each side is continually reacting to the actions of the other side and to the state of the environment while attempting to complete the mission, air combat can be described in terms of agent behaviors.

However, the model structure is mixed, as it also can be described in terms of the equations governing how aircraft fly and weapons deploy. Therefore, a mixed ABM and equation-based model is best as a system representation of the air combat system. We feel that a simulation model that combines ABM with equation-based representation of weapons and platforms provides a valid representation of air combat as it includes not only the mechanical functioning of the weapons and vehicles but also a model of the

behaviors associated with the pilots as they assess each situation and decide on an appropriate course of action.

**Agent Decision Making in Simulations.**

As stated previously, agent based modeling uses artificial intelligence techniques from the computer science word to simulate agent decision making. Gat (1998) talks about a three-layered approach to robotic control systems. Gat's (1998) three layers in the AI control structure are the "deliberative", "reactive", and the "sequencing" layers. This structure seems to be common to robotic control systems, as Gat points out. Although Gat and others developed this three-layer architecture for robotic control systems, many AI computer programs have taken advantage of the ideas behind the three-layer structure.

The deliberative layer is the planner. The job of the deliberative layer is to plan intermediate goals on the way to achieving the overall goals of the agent. This layer can be thought of, within the air combat context, as the higher-level cognitive functions of the pilot that plan ahead and attempt to match weapons to targets, set waypoints, determine optimal flying formations, and other planning functions.

The reactive layer is the set of behaviors that responds to the environment. The reactive layer chooses the best actions to cope with the state of the world as it is. For instance, a pilot may suddenly have an enemy missile lock on his aircraft and need to conduct evasive maneuvers. The reactive layer is a composed of base behaviors and actions that have no memory of the state of the world. It is fully vested in the current state and the best reaction to that state.

The sequencing layer is short-term goal driven. The sequencing layer is the layer that allows the reactive layer and the deliberative layer to work together to accomplish goals. Based on goals received from the deliberative layer, the sequencing layer sequences reactive actions to react to the environment while still striving to meet goals. The sequencer has memory of the state of the world only in that it will remember actions taken that may influence future actions. For instance, if an action taken was not successful in, say, destroying an enemy target, the sequencing layer “remembers” that the last action taken was not successful. The layer then checks if destroying the enemy target is still a goal and queries the reactive layer for an action different from the last unsuccessful action to use for accomplishing the goal of destroying the enemy target.

**Agent Deliberative Planning Functions.**

In context of a mission level combat model, the agents must be able to conduct three main functions within their deliberative planning layer. First, the agent plans routes and sets waypoints to meet overall mission goals. In a movement to contact, the mission would be to clear some battle space by attempting to gain contact with enemy agents. To plan a route through this battle space with this goal, some sort of algorithm for efficiently searching throughout the space could be used, keeping track of where the agent has been and then planning the next waypoint. There are numerous heuristic and analytic methods for finding optimal routes depending on the mission. Our research focuses on a single scenario with fixed routes, so this planning function is not needed.

The next two functions relate to engaging an enemy agent. The agent’s deliberative planner must be able to decide what tactics to use when engaging the threat and the agent must assign specific weapons to specific targets. Choosing tactics can

simply be a rule-based heuristic depending on the situation, the enemy weapons and capabilities, whether the enemy agent is aware of the agent or not, and many more considerations.

The weapon-target assignment problem, on the other hand, has been extensively studied. Assigning a weapon to a target while accounting for amount and types of weapons available, type of target, possibility of future targets, and many other factors, can quickly become a computationally intensive task. Many methods have been proposed for solving this problem, mostly as a matter of optimizing the outcomes over a period of time. Ahner and Parson (2013) propose a dynamic programming approach that uses Monte Carlo methods and a Markov Decision Process-like algorithm that would solve the problem for the simulation scenario, create an optimal policy, and then provide the agent the optimal policy as a tool for execution within the simulation. Genetic and other Evolutionary methods have been proposed for finding the optimal solution to this problem (Hill, et al. 2001; Chen, et al. 2009). Even game theory has been proposed as a method for use in solving these types of problems for an agent in a simulation (Cruz, et al., 2001). More discussion describing various linear programming, network flow, and heuristic algorithms for solution of the weapon-target assignment problem can be found in the paper by Ahuja, et al. (2003).

The method we focus on in this research for assigning a weapon to a target is to evaluate each target in the context of the simulation time with a simple heuristic. The flight lead agent considers the range to the target, number and types of weapons left and their effectiveness against the particular target, the perception of what the target's lethality is against an agent's own platform, and the probability of targets existing in the

future of the simulation. Once the algorithm finds that for each target, the agent assigns weapons/assets to each target according to some value rule. This method can be considered a greedy approach that may not yield an optimum weapon-target pairing. However, the point of an AI in a simulation is to provide more "human"-like decision making to bring the model more in line with real world decision making.

**Agent Reactive Behavior Architectures.**

There have been many proposals for reactive behavior architectures from both the world of robotic control and video game AI. Many of them have their roots in the combined simple behavior machines of Braitenberg (1984). These machines combined basic, very simple behaviors that, when combined, produce complex, unexpected behaviors that could be likened to human behaviors. Finite state machines provide a way for an agent to constantly sense the state of the world and then react by moving to a different state if the sensed state of the world signals the need to change based on a set of defined transition rules. Spronck, et al. (2006) propose one particularly interesting reactive architecture in which the agent is assigned a randomly generated set of rules from a master list for reacting to different situations. When the agent encounters an engagement with enemy agents, a weighting algorithm updates the agent's rules based on how well they worked in the engagement. In this manner, the pool of probable rules to be included in the AI's engagement script is optimized. This is a type of reinforcement learning algorithm (Spronck, Ponsen, Sprinkhuizen-Kuyper, & Postma, 2006).

One of the most common reactive architectures used for agent AI's is Behavior Trees (BT). BTs have been widely used in video gaming in games such as Halo 4 to make the game AI's more dynamic and provide a more realistic experience to the player.

Marzinotto, et al. (2014) provides a very good discussion of BTs and their construction, including a mathematical basis and comparison with finite state machines.

BTs are rule sets that operate by attempting to provide execution instructions to the agent based on the state of the simulation at the time the agent queries the BT root node. Execution of the BT begins at the root node. Each BT has only one root node, but there can be several different BTs even within the same agent used for different situations. The root node queries down the tree and each sub-node attempts to execute its subordinate behaviors according to each sub-node's basic type. There are usually four non-leaf nodes cited in literature (Marzinotto, Colledanchise, Smith, & Ogren, 2014). The four are Selector, Sequence, Parallel, and Decorator nodes. The symbols used for Root, Select, and Sequence throughout this study are shown in Figure 2 through Figure 4.

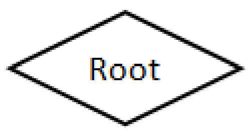



**Figure 2: Root Node (Standard Symbol Adopted throughout this paper)**

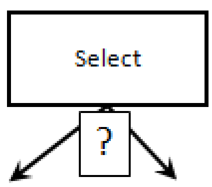



**Figure 3: Selector Node (Standard Symbol Adopted throughout this paper)**

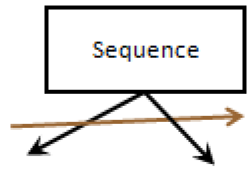



**Figure 4: Sequence Node (Standard Symbol Adopted throughout this paper)**

Selector nodes attempt to run subordinate nodes from left to right until they find a node that will run. This means that the first node the selector encounters that is capable of executing is the only node that the selector runs. A sequence node attempts to run each of its subordinate nodes until it hits a subordinate node that does not run. If the sequence of subordinate nodes does not complete, the sequence node will return a false, because the sequence node was unsuccessful. A parallel node, of which the root node is a special case, attempts to run all subordinate nodes and behaviors simultaneously. Decorator nodes control the synchronization of separate agents with different behavior trees. In other words, this type of node allows cooperative behaviors between different agents (Marzinotto, Colledanchise, Smith, & Ogren, 2014). This study does not use decorator nodes explicitly in because AFSIM does not yet have resources in the scripting language that allow this type of node. Instead, between agent cooperation is somewhat hardcoded into the BTs for each agent, as we will discuss in more depth in Chapter 3.

Leaf nodes are the very basic node at which execution of the behaviors takes place. These nodes have no subordinate nodes. The two types of leaf nodes are Action and Condition (Marzinotto, Colledanchise, Smith, & Ogren, 2014). Figure 5 and Figure 6, respectively, illustrate the standard symbols we have adopted within this paper for representing these two types of nodes. Action nodes execute actions. These nodes use

algorithms to calculate speeds, trajectories, weapons release points, etc, and then make the agent perform turns, increase/decrease in speed, etc. Conditional nodes test the state of the agent's environment (the simulation) for some condition. Conditional nodes are usually combined with a set of action nodes under a sequential node.

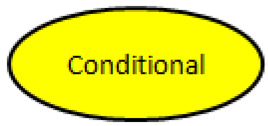



**Figure 5: Conditional Node (Standard Symbol Adopted throughout this paper)**

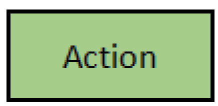



**Figure 6: Action Node (Standard Symbol Adopted throughout this paper)**

Figure 7 depicts a very basic BT by way of example. This example was adapted from Marzinotto, et al. (2014), to illustrate the operation of a simple BT. The example is a BT for a robot, but could easily be applied to a simulation of a robot where the robot is the agent in the simulation. The robot has a goal to walk forward. Execution of the BT begins at the Root node. The Root node simultaneously attempts to run all subordinate nodes. In this case, there is just one, a Select node. The Select node starts with the subordinate node furthest to the left, which happens to be another Select node. This subordinate Select node runs a sequence that checks if the motor is too hot to run or has a low battery. If this Sequence node is unable to run, then the node passes a false back to

the parent Select node. The parent Select node then attempts to run its next subordinate node, another Sequence node that checks if the robot has fallen down.

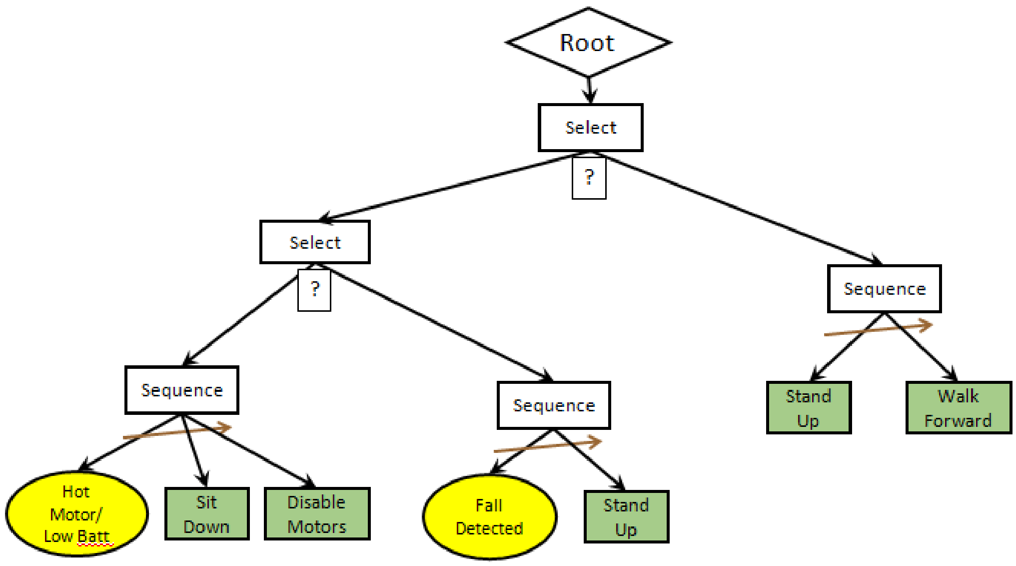



**Figure 7: Example Behavior Tree layout (Adapted from Marzinotto, et al.)**

If neither of the Sequence nodes returns running or true, then the parent Select node returns false. This causes the Select node just below the Root node to move on to its next subordinate node, a Sequence node. This next Sequence node executes a "Stand Up" behavior (Action node). If the robot is already standing, the stand up behavior will just return "running". Running or true tells the Sequence node to move to the next subordinate behavior, which is a "Walk Forward" behavior. Note that this BT could become much more complicated if it accounted for navigational instructions, obstacle avoidance, or other interactions with the environment such as picking up and dropping objects.

Another reactive behavior architecture proposed by Woolley and Peterson (2009) is called the Unified Behavior Framework (UBF). The UBF has a tree data structure similar to BTs, but they differ from BT's in two main ways. First, the entire UBF evaluates before the agent executes any actions. The UBF returns a recommended set of actions to the agent and the agent then implements this recommendation. This separates out the "thinking" from the "doing". Secondly, arbiters at each level of the tree evaluate the child behaviors recommended actions using both the magnitude of the actions proposed and the value of the vote given by the behavior. Each arbiter is essentially a heuristic or value function that chooses which actions to implement at that level of the UBF tree. There are many different types of arbitration algorithms used, some of which Woolley and Peterson (2009) detail.

The UBF works by populating a vector of basic actions, for instance heading, altitude, thrust, fire weapon, etc., at the root node of the UBF tree. This action vector comes straight from the arbiter at the top of the tree. The UBF then passes this vector to the agent for execution. The theory is that this way of combining very basic behaviors into an action vector can lead to some emergent behaviors as the agent deals with the state of its environment (Woolley & Peterson, 2009). Figure 8 shows an example of the UBF structure with the action vector output from the top node of the UBF tree. The controller represents the sequencing layer of the agent.

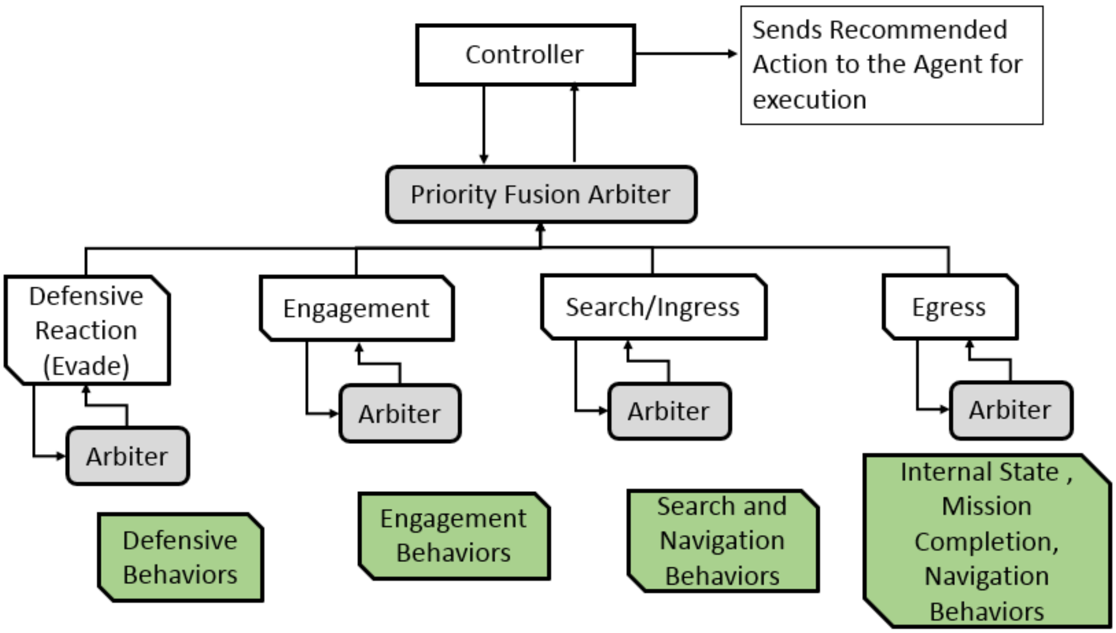



**Figure 8: Unified Behavior Framework Example for an agent conducting air combat**

**Sequencing Behaviors and incorporating mission goals.**

The sequencing layer manages agent behaviors and accounts for near, mid, and long-term goals in agent decision making. The sequencing layer has the ability to "remember" the state of actions past. For example, if a robot tried to turn left after encountering an obstacle, but encountered another obstacle, the sequencing layer would have memory of this action and not allow the reactive layer to place the robot back into the state where it is facing the original obstacle.

In addition, it receives goals from the deliberative layer. It queries the reactive layer for action by feeding it those goals necessary for the reactive architecture to operate, such as the task of destroying a detected enemy and the state of that enemy. The

reactive architecture than calculates the behavior needed to "react" to that "state", for instance conduct a pure pursuit (head on) and fire weapon.

The sequencing layer performs agent cooperation, such as targeting, formation flying, and tactical cooperation, in order to achieve near-term and end game goals. In BTs, the sequencing layer has to be thoughtfully incorporated into the tree structure in order to work properly. In other words, part of the sequencing behavior is the order in which the Select and Sequence nodes encounter subordinate nodes. The deliberative layer incorporates another part of the sequencing layer along with the resources that allow communication of the simulation environmental state to the BT.

**Analytic Framework for Simulation (AFSIM).**

AFSIM, formerly known as Analytic Framework for Network Enabled Systems (AFNES), is an agent-based simulation framework developed by Boeing and now managed by AFRL/RQ. A simulation framework, like AFSIM, is a set of tools, also known in the programming world as a library, and is used for loading simulation scenarios, populating different objects within the simulation, and then controlling the simulation execution (Zeh & Birkmire, 2014). Because AFSIM is object-oriented, it is important to define here what we mean by "objects" in the context of AFSIM and simulation, in general. Objects can be almost anything within a program. Platforms, sensors, and weapons are examples of objects that populated within AFSIM. Figure 9 provides a depiction from the AFSIM Overview Report of all the simulation control components and simulation objects that reside within a scenario in AFSIM (Zeh & Birkmire, 2014). AFSIM uses a special simulation scripting language to define objects. Agents in AFSIM are then really a combination of different platform, sensor, and weapon

objects. The heart of an agent is the decision making and information flow produced by the processor objects. Some of these processors are discussed in more detail in Chapter 3, but are briefly mentioned in the rest of this section.

AFSIM uses a base simulation engine, called Simulation of Autonomously Generated Entities (SAGE), and the framework has the ability to add different models into the framework as plug-ins. SAGE reads in the text files defining the simulation scenario and executes the AFSIM commands in the text files by calculating interactions between the defined objects over time in a discrete-event manner within the context of a specified geographical area. AFSIM also includes an agent behavior engine, called the Reactive Integrated Planning aRchitecture (RIPR) which implements a Behavior Tree reactive behavior architecture coupled with “quantum-tasker processor” objects that act as the deliberative and sequencing layers of the AI architecture. The RIPR model is discussed in more detail in Chapter 3.

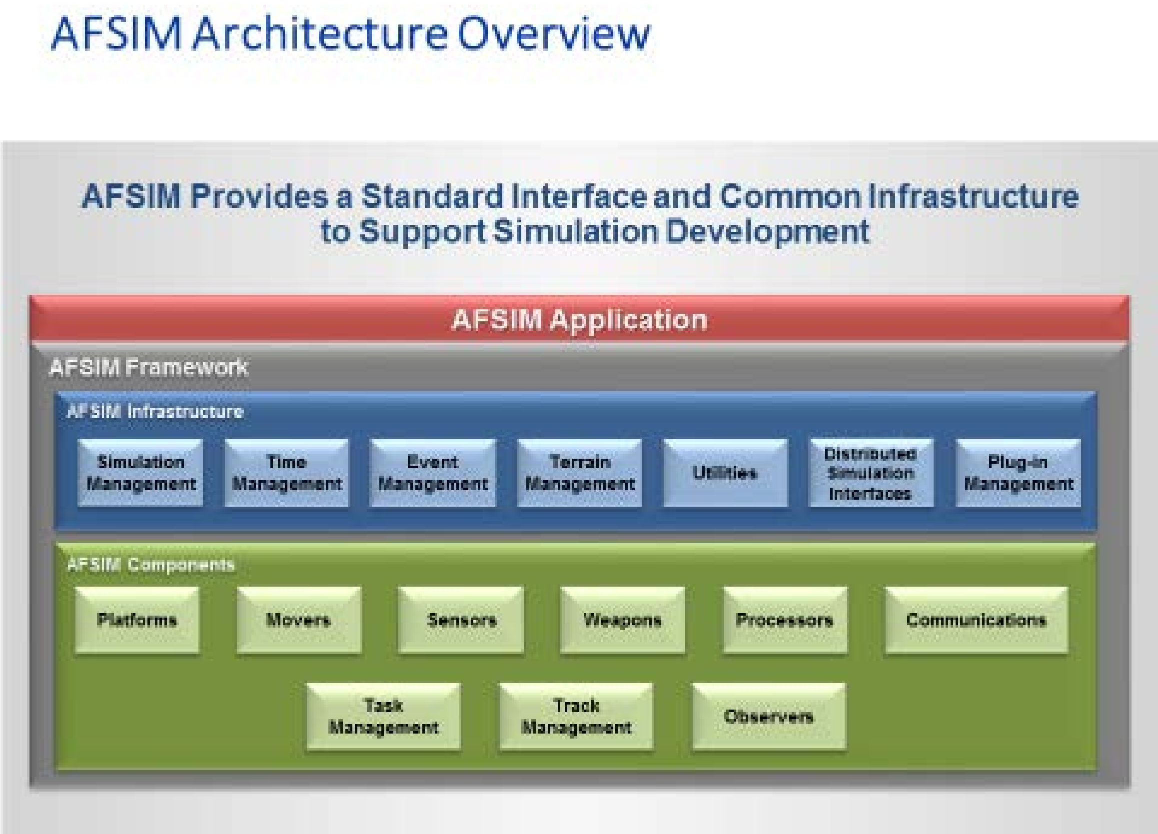



**Figure 9: AFSIM Architecture overview showing the simulation control infrastructure and simulation components (Zeh & Birkmire, 2014)**

Objects that make up an AFSIM scenario include platforms (ground vehicles, aircraft, missiles, etc.), sensors, communications systems, weapons, processors used to perform calculations on tracks or make decisions, scenario definition, input/output objects defining setup files and output files, and other script objects defined by the user that contain AFSIM commands. The scripting language is a C++ like programming language that allows access to AFSIM library objects. Figure 10 depicts the various objects that make up a platform within AFSIM and can be accessed through the scripting language (Zeh & Birkmire, 2014).

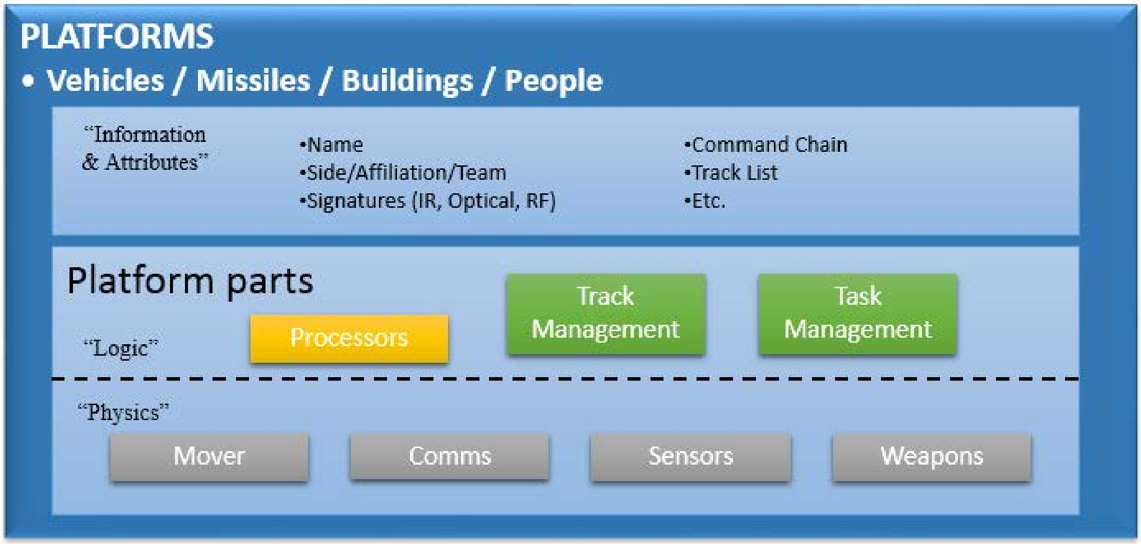



**Figure 10: AFSIM Platform Components (AFSIM Overview, 2014)**

## Analysis of Weapon Systems Using Simulation

### Statistical Approaches to Analyzing Simulations.

There are several statistical approaches for analyzing simulation model output in relation to the main factors. A survey of some of the recommended methods is given in the next section. We detail statistical methods for conducting a designed simulation experiment in the section after that.

### Traditional Simulation Scenario Analysis.

Most traditional statistical techniques for analyzing outputs of a simulation revolve around comparing one or more simulation scenarios to each other. An example is to compare the mean number of hits by a specific weapon for one scenario with one weapon system to the mean for another scenario with a different weapon. Several statistical measures of comparison are used.

First, we discuss some commonly used techniques for the comparison of two scenario outputs. The means of the output data for two scenarios can be compared using the Paired t-Test or the Two-Sample t-Test assuming unequal variance (Welch T). Additionally, the medians of the data can be compared using the non-parametric Wilcoxon Rank Sum Test.

The Two-Sample t-Test assuming unequal variance requires independent input samples, and that the response (output data) is normally distributed. With most outputs of a simulation, the outputs are assumed normally distributed, but not always. This test is robust to the normal assumption, so small problems with this particular assumption will not cause large issues. This test is often used when the analyst has different sample sizes for each of the input systems and cannot assume equal variance of the two populations. A good discussion of the modified two-sample t-Test is in Law (2007).

One important thing to note is that several sources on simulation analysis prefer confidence intervals to a hypothesis test and a p-Value approach (Law, 2007). The main reason is that confidence intervals provide more data in terms of the magnitude of a difference. The hypothesis test and p-Value give no indication of "how" significantly different the two populations are. For instance, if the difference in means of the two samples is statistically significant but only 0.02, is this difference truly, practically significant? In addition, the analyst must use the p-Value carefully, as it has a higher probability of showing a significance when there is none (Nuzzo, 2014).

The next statistical comparison method is the Paired t-Test. The paired-T test is always safe to use when comparing two normally distributed system responses. There is still a normality assumption, and the sample sizes must be equal, but there is no longer

any assumption about the variance of the two sample populations required. The Paired t-Test is useful for comparing highly correlated data.

One particularly useful technique for use in conjunction with the paired t-Test is Common Random Numbers. Many systems are subject to a large amount of noise, or variability due to factors outside the control of the simulation analyst. When a model exhibits a large amount of variance, one way to help reduce the variance and thereby make the signal in the output response more visible to the statistical tests is to induce correlation in the simulation between scenarios/systems, and then use the paired-T test to analyze the output. Using common random numbers (i.e., the same stream of random number seeds to generate random numbers within each scenario) may often induce positive correlation between the two models and reduce variance in the outputs. The paired-T test must be used with correlated data like this. A comprehensive discussion on the topic may be found in Law (2007) and in Banks, Carson, Nelson, and Nicol (2004). Note that for this technique to be the most effective, the CRN must be synchronized between each scenario. This means that for every random number draw for the same event in each scenario, the random number draw must be on the same random number stream. Unsynchronized CRNs may induce some correlation, but not as much as fully synchronized random number streams. Again, for the paired-T test, and for all the statistical tests that account for variability, confidence intervals are still preferred over the p-Value approach.

Another comparison method for two samples from two populations is the non-parametric Wilcoxon Rank Sum test. This method is useful because it measures the spread of two populations from each other. One advantage of this is that the analyst does

not have to assume a normally distributed response for either population. In fact, the Wilcoxon Rank Sum works on any distribution of the tested samples. Because of this, no assumptions about the variance is necessary. The only assumption needed to use this method is that the two samples come from similarly distributed populations. The interested reader can find more on the Wilcoxon Rank Sum test in Wild and Seber (1999).

One drawback for using the paired t-test, the two-sample t-test, or the Wilcoxon Rank Sum is that they can only be used to compare two samples without making modifications to the tests. One way to conduct a t-test comparison between several alternative scenarios is to use all pairwise t-tests. To do so, the most appropriate comparison involves using some method to stabilize the overall confidence level. If one constructs confidence intervals for all pairs of alternative scenarios without adjusting the confidence level for each simultaneous comparison, then the overall confidence level will be incorrect. One method uses the Bonferroni inequality to adjust the individual confidence levels of each pair of simultaneous comparisons. The procedure is straightforward; simply divide the desired overall stated confidence level, say 0.05, by the number of confidence intervals, c, to get the individual confidence level of each comparison. One can see immediately that this will result in lower confidence of all the individual comparisons, with wider intervals. This method has a reduced power to see differences between each of the alternatives. Law (2007) has a good discussion on this method and other pairwise methods for defining a simultaneous confidence level for comparing multiple alternatives.

A modification of this all-pairwise simultaneous comparison method for multiple alternative samples is to compare all alternatives with a standard. This modification reduces the number of pairwise simultaneous confidence intervals requiring construction, thereby tightening the interval widths themselves and allowing more power to see significance. To use this method, the analyst is required to identify one of the alternatives to use as the "standard".

Several other important methods for comparing two or more alternative scenarios/systems are available. The Two-Stage Bonferroni procedure for comparing two or more alternatives uses a t statistic and an estimated variance for each system response to calculate confidence intervals of a specified precision in order to directly compare the means of the two systems. This is a two-stage procedure in which the analyst specifies the precision (error,$\varepsilon$), initial run number,$R_0 > 10$, and the probability of correct selection (PCS), $1 - \alpha$, in the first stage. The analyst makes the required $R_0$ simulation runs, estimates the variance from this initial sample of each system, and then uses this estimate of variance to establish a required minimum number of runs, $R,$ to reach the specified precision for each alternative scenario. The analyst directly compares the means of each alternative scenario's sample after completing the additional $R - R_0$ runs. An excellent discussion of this method is in Banks, Carson, Nelson, and Nicol (2004).

Another method comparing multiple scenarios is a non-parametric ranking and selection method that makes use of the Multinomial Selection Problem. A good discussion of this method is given in Bechhofer, Elmaghraby, and Morse (1958) and an alternative version of the method in Miller and Nelson (1996). The method is capable of

detecting a significant difference between two or more systems at a specified "zone of indifference", which can be thought of as a "practically significant difference" between the two populations (Miller & Nelson, 1996). One drawback is that it does not detect the magnitude of the difference between the alternative systems; it can only identify the best. This method would be a good procedure to use for identifying the best tactical strategy of several, where magnitude of difference may not be as important, but the fact that the strategy increases the odds of success is important.

**Designed Experiment (DOE) Approach to Simulation Analysis.**

Designed experiments are conducted to maximize the amount of information about a system obtained through a minimum number of runs. The design provides us with appropriate statistical analysis tools that can be used to provide insight into the dynamics of the given system. The principals of randomization, replication, and blocking (local control of error) drive the overall design of experiments (DoE) philosophy (Montgomery, 2009). Randomization involves randomizing the run order of each combination of levels of the factors (treatment). Replication is the repeated measurement of a particular treatment. Blocking is way to control nuisance factors that introduce error into the system response measurements. Another major part of the DoE philosophy is sequential experimentation. This principal says that the experimenter should not use all experimentation resources in the first experiment, but rather use a fraction of the resources and then use the results of the first experiment to inform further experimentation (Montgomery, 2009).

One-factor-at-a-Time (OFAT) experiments are those in which all the factors are held at a constant level while one factor's levels are varied at any given time. This type

of experimentation gives us information about the main effects of the factors, but does not provide information on possible interactions or higher-order (non-linear) effects of the factors. It is not a very efficient use of experimentation (Montgomery, 2009).

A common type of design is the $k$-factor, 2-level, factorial design. This design is a cubic design that provides runs (or design points) at each of the two settings (low and high) for each of the factors. This design type is called the $2^k$ factorial experiment. Full factorial designs incorporate a run at every combination of every level of each of the factors. Figure 11 shows a cubic plot of some design points for both a $2^2$ factorial (a) and a $2^3$ factorial design (b), wherein each dot at each corner point represents one run. The design points allow estimation of the effects (what happens to the system) as each factor moves from low to high (or vice versa) and interaction effects between the factors. An experiment with $k$-factors at 2-levels each has $2^k$ number of runs for one replication (for example, a $2^2$ has 2 x 2 = 4 runs of one replication of the design.)

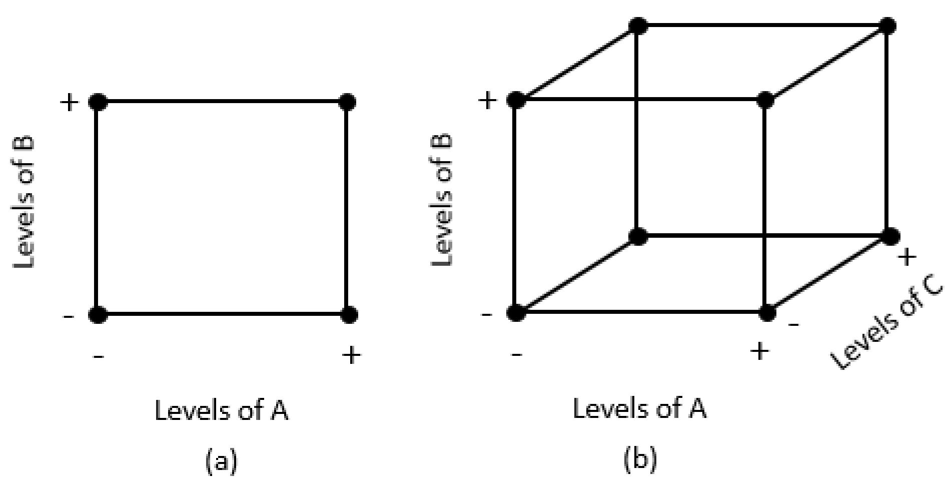



**Figure 11: Design matrix for a (a) 2 factor 2 level factorial design and (b) a 3 factor 2 level factorial design**

A full factorial experiment's number of design points (or the number of runs required for a single replication) grows exponentially as the number of factors involved grows. Fractional factorial designs based on $2^k$ full factorial design can be implemented which cut the number of runs required by a certain fraction. Fractional factorial experiments, or $2^{k-p}$ fractional factorial, also have a loss of information associated with them due to the loss of design points. These designs can be very useful in screening, blocking out noise factors, and can be folded over (adding runs to the original design) to create full factorial designs when conducting sequential experimentation. Both the full factorial and fractional designs are analyzed using analysis of variance (ANOVA) methods. This analysis examines the variance structure to find which factors are important in explaining the response. An in-depth discussion of $2^k$ full factorial and $2^{k-p}$ fractional factorial experiments may be found in (Montgomery, 2009).

A $2^k$ factorial design allows estimation of a first order with interaction model of the form,

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i<j=2}^{k} \sum \beta_{ij} x_i x_j + \varepsilon \qquad \textbf{(1)}$$

Curvature effects are estimated in order to test if there is significant curvature in the system by adding center runs and calculating the single-degree-of-freedom sum of squares for pure quadratic curvature given by

$$SS_{Pure\ Quadratic} = \frac{n_F n_C \left(\overline{y}_F - \overline{y}_C\right)^2}{n_F + n_C} \qquad \textbf{(2)}$$

Where $n_F$ is the number of factorial design points and $n_C$ is the number of center runs conducted. Note that this method is only appropriate for continuous factors. There are

methods available for estimating curvature in a model of mixed continuous and categorical factors as long as at least one of the factors is continuous (Montgomery, 2009).

If there is significant curvature in the system, then a response surface should be estimated using a second-order or higher model. A good design for estimating the second-order model is the Central Composite Design (CCD). The CCD is created by simply augmenting a $2^k$ factorial design with axial runs at some distance, $\alpha$, from the center of the design, as shown in Figure 12.

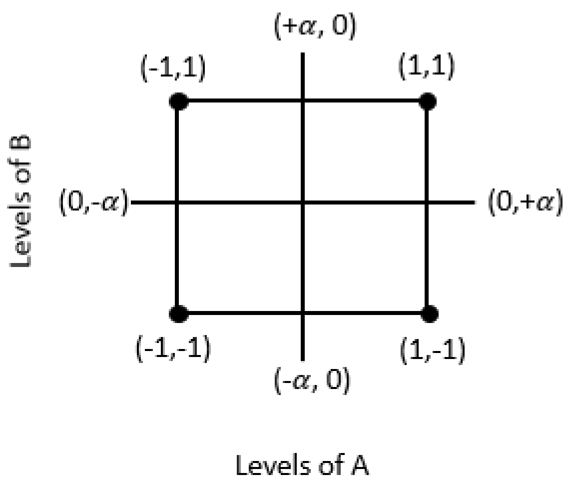



**Figure 12: Augmentation of the $2^k$ design with axial runs to form a CCD**

This allows estimation of the second-order model of the form,

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i<j=2}^{k} \sum \beta_{ij} x_i x_j + \varepsilon \qquad \textbf{(3)}$$

There are many other designs for estimating the response surface including the Box-Behnken 3-level designs, equiradial designs, Hoke designs, Koshal designs, Hybrid

designs, D-, G-, and I-optimal computer generated designs, and Small Composite designs. A good discussion of these designs can be found in (Myers, Montgomery, & Anderson-Cook, 2009).

Mixture Experiments are response surface experiments where the factors are components of a mixture. For instance, if a combat aircraft uses a mix of missile types, say a medium-range missile and two types of short-range missiles, there are three mixture factors, as there can only be so many total missiles carried by the fighter. Mixture experiments are special in that they involve the proportions of each of the component factors in the mix. A special coordinate system, the simplex coordinate system, is used in these experiments instead of the standard cubic coordinate system because the sum of the proportions of the three components always has to add to one. A diagram of the simplex coordinate system for three components is shown at Figure 13. A four-component simplex is a pyramid.

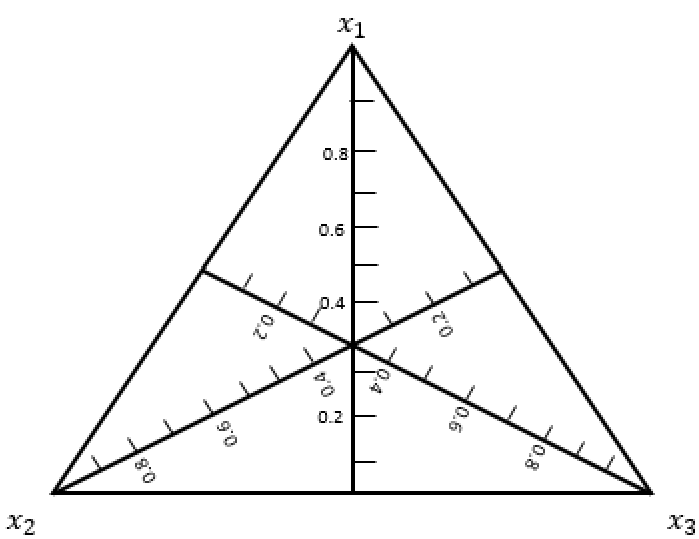



**Figure 13: Simplex Coordinate System for three components**

As shown, any point on the system in $x_1, x_2, x_3$ has a set of coordinates that result in a sum of one. There are several designs for mixtures making use of the simplex coordinate system. Simplex Lattice designs are uniformly spaced points on the simplex. Simplex-Centroid designs have points located at the centroids of the simplices. These designs can be augmented with axial points to provide better estimation of higher order models. Analysis of the response surfaces is conducted using the usual ANOVA techniques; however, there are several non-standard models of the response surface, called the Scheffé polynomials. There are Linear, Quadratic, Full Cubic, and Special Cubic forms of the polynomial. For more discussion of mixture designs and their analysis, see (Myers, Montgomery, & Anderson-Cook, 2009).

If there are process variables (variables that have effects on the mixture but are not a component of the mixture) within the system, for instance a chemical mixture affected by temperature and pressure, there are two approaches to experimental design. The first is to transform the mixture factors into $q - 1$ independent variables, through use of ratios, and then perform a standard experiment, such as a factorial design. The second is to perform Simplex type experiments at the different levels of the non-mixture process variables. In other words, perform a simplex experiment in the mixture variables at each design point of a factorial experiment in the non-mixture variables, or perform a factorial experiment in the non-mixture variables at each design point of the simplex experiment of the mixture variables (Myers, Montgomery, & Anderson-Cook, 2009).

The last class of experimental designs considered for use in this study is computer-generated designs. These designs are constructed using algorithms to optimize the variance structure of the design. D-optimal design criteria concentrate on minimizing

the variance of the estimated model coefficients.  G-optimal designs focus on minimizing the maximum prediction variance of the estimated model.  I-optimal designs minimize the average prediction variance of the estimated model.  JMP has some very useful algorithms for creating D-, G-, and I-optimal designs.  However, care must be taken in using computer-generated designs to ensure that the analyst understands the design being output by the software.  Several different designs and their variance structures should be compared before choosing a particular design for implementation.  A good discussion of computer-generated designs is found in Myers, et al. (2009).

**Summary**

In this chapter, we surveyed literature to build a toolbox of useful methods in helping us construct a methodology for analyzing a new missile system using an agent-based model.  Simulation and modeling in the DOD walks through the types of simulation models and several sources help us better understand the level of aggregation and resolution required for a combat model used to analyze weapon systems.  We show that agent-based simulation models are ideal for use in analysis of complex adaptive systems, such as air combat.  There are several different potential architectures from AI theory to provide model of agent decision making within an ABM.

AFSIM, a mission-level model ABM framework, is the tool of choice for this research.  For the scope of the problem introduced in Chapter 1, a mission level model is appropriate.  If strategic effects of using the new weapon system are sought, several potential models could be used at the campaign level.  However, we caution that analysis of the tactical effects of a weapon system should be determined within several different

scenarios with higher resolution models before seeking to learn what strategic effects exist. This is primarily because the higher resolution model will yield insights directly applicable to development of lower resolution campaign models.

Finally, we surveyed several different simulation output statistical analysis techniques, including statistical comparison techniques and design of experiments. As is discussed in Chapter 3, we choose to use the statistical comparison techniques for model verification purposes and use the DoE techniques to extract more information about factor effects on the air combat system.

# III. Methodology

## Overview

The general methodology is depicted in Figure 14. The first step is to define the metrics used within the study and collect data. A number of sources are used for collecting weapons data and advice on appropriate metrics including subject matter experts from AFRL and Lockheed-Martin, Jane's Defense Catalogues (Hewson, 2009) (Jackson, Munson, Peacock, Bushell, & Willis, 2013) for different weapons systems types, and various doctrinal publications and air combat tactics studies.

Using a designed experiment, the base simulation scenario is modified for each specific combination of the levels of the factors being studied. Once the simulation models are constructed, they are verified using graphical techniques and subject matter experts from AFRL and Lockheed-Martin. Validation of the underlying models is an ongoing effort by AF/A9 and AFRL and is briefly discussed in the Section 3.4.

Finally, the AFSIM simulation runs are conducted, output data parsed and collected and then analyzed using the simulation analysis methodologies discussed in section 3.5.

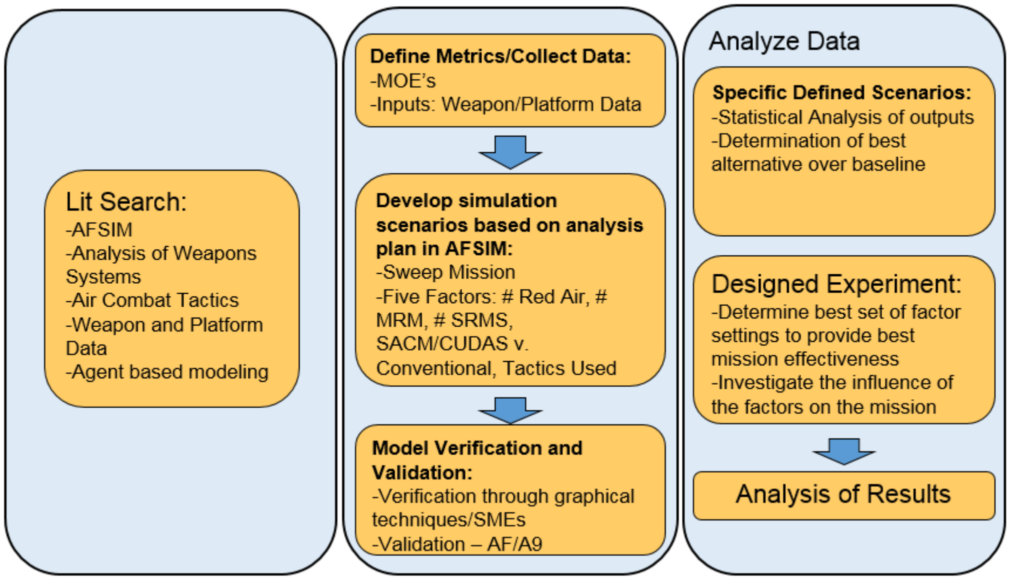



**Figure 14: Simulation Study Methodology for the Weapon System Analysis**

### Metrics Definition and Data Collection

The analysis plan calls for isolating the effects due to the new weapon; therefore, many of the factors influencing air combat are held constant.  For instance, all air platforms use the same generic fire control sensor and have the same maneuverability characteristics based on F-15-like fourth generation fighter aircraft.  The main factors that are allowed to vary across simulation scenarios are the following:

1. Number of Medium Range Air-to-air missiles loaded on Blue platforms at mission start.
2. Number of Short Range Air-to-air missiles loaded on Blue platforms at mission start.
3. Number of the SACM/CUDAS-like new weapon.
4. Number of Red Fighters within the scenario.
5. Tactics used by Blue flight during air-to-air engagements

We use the first three factors to investigate different mixes of weapons and their effects on the outcomes of a sweep-type mission. The second and third factors show the differences between using the current AIM-9X type weapon and using the new agile type of weapon. In the various scenario modifications, different mixes of the current medium range missile and the SACM/CUDAS-like new weapon are investigated, from no SACM-all MRM to no MRM-all SACM initial mission loads. Two different replacement ratios are investigated: replacing every one MRM/SRM with two CUDAS-like missiles (2:1 ratio); replacing every one MRM/SRM with three CUDAS-like missiles (3:1 ratio). Table 1 shows the carrying capacities for the different types of weapons on an F-15E. The chosen scenario uses a fighter platform that is similar to the F-15E (fourth generation fighter aircraft) for both the Blue and Red sides. Data for the fourth generation fighter model comes from Jane's All the World's Aircraft (Jackson, Munson, Peacock, Bushell, & Willis, 2013). From Table 1, it is immediately apparent that no combination of the weapons will exceed the maximum weapon load weight. For the simulation, the fighters will have the capacity to carry eight MRM/SRM or the appropriate number of CUDAS missiles shown if the entire load out is only one type of air-to-air missile. Additionally, the Blue fighters always carry two air-to-ground weapons. The MRM is modeled as an AIM-120 AMRAAM; the SRM as an AIM-9X Sidewinder; and the air-to-ground weapon is modeled as the GBU-38,500lb variant JDAM. Note that each of these weapons is not modeled exactly, due to classification issues, but rather is modeled based on open-source material found in Jane's Air Launched Weapons (Hewson, 2009).

**Table 1: Weapon Carrying Capacities for the F-15E-like platform**

| | Kilograms | Pounds | |
|---|---|---|---|
| Max. Wpn Load | 11113 | 24500 | |
| | MRM/SRM | CUDAS (2:1 Ratio) | CUDAS (3:1 ratio) |
| Number Wpn Pylons | 8 | 16 | 24 |

| Weapon | Weight (kg) | Proportion of Max. | Max Number |
|---|---|---|---|
| Medium Range | 161 | 0.01449 | 69 |
| Short Range | 85 | 0.00765 | 130 |
| CUDAS-like | 49 | 0.00441 | 226 |
| JDAM-like | 227 | 0.02043 | 48 |

We vary the number of Red aircraft that Blue must face in order to investigate how the Blue systems work under different difficulty levels, easy to hard. The number of Red aircraft is varied from four, six, to eight total aircraft flying CAP missions in pairs at different locations along the Blue attack route. All Red aircraft have the same fire control sensors and maneuverability as the Blue aircraft in order to isolate as much as possible the effects on tactics and mission success due strictly to the introduction of the new missile system. Red aircraft are only armed with a standard combat air patrol weapon load and do not have their weapons varied over the scenarios.

Blue tactics are investigated by changing the range of tactics available to the pilots (agents) from scenario to scenario. As discussed previously, there are generally two phases of tactics in air combat, Beyond Visual Range (BVR) tactics, and Within Visual Range (WVR) tactics. BVR tactics tend to influence WVR tactics (Baker, 1986). BVR tactics available to use before the merge are single-side offset, straight-in, lead/trail, and pincer. The single-side offset tactic, shown in Figure 15, involves the flight lead

moving the entire flight off-axis of the approaching enemy aircraft, usually up to 30 degrees. This gives the flight the ability to attack at a more oblique angle to the incoming enemy.

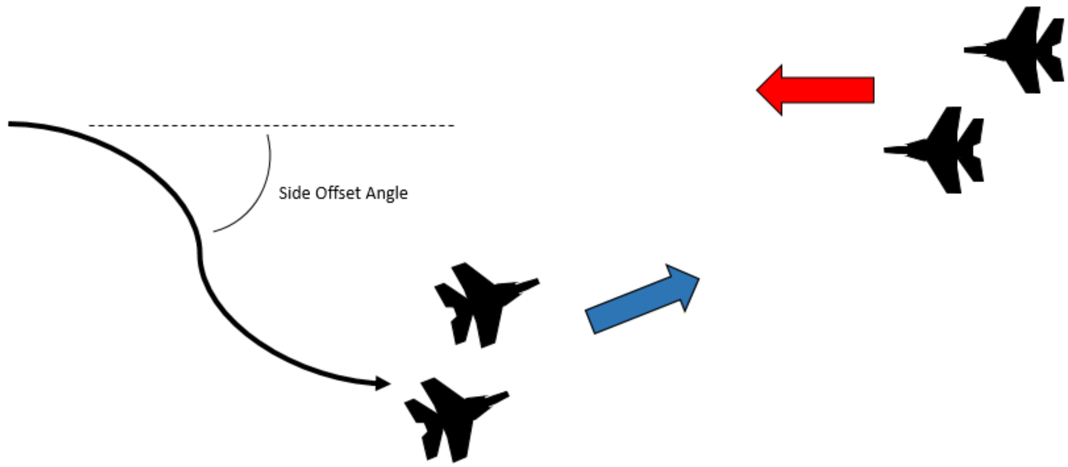



**Figure 15: Single Side Offset Maneuver at BVR**

Straight-in is a tactic, Figure 16, which means the flight flies straight at the incoming enemy. A flight usually uses this tactic when the flight lead feels that friendly weapons and fighters have superior range and maneuverability against the perceived enemy fighter/weapon types.

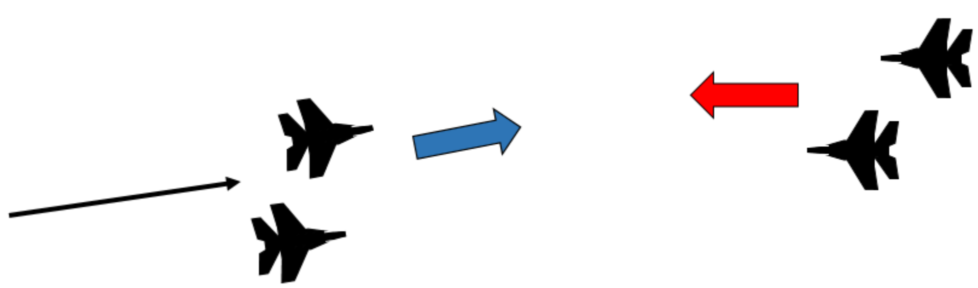


**Figure 16: Straight-In maneuver at BVR**

A flight uses the Lead/Trail tactic to avoid merging, or coming to WVR, as long as possible. One pair of the Blue flight moves straight-in and fires weapons at BVR, while the other pair loops backwards to maintain standoff and then fires their weapons at BVR while the first set of Blue fighters conduct the loop back. The Lead/Trail maneuver is illustrated with a flight of two in Figure 17.

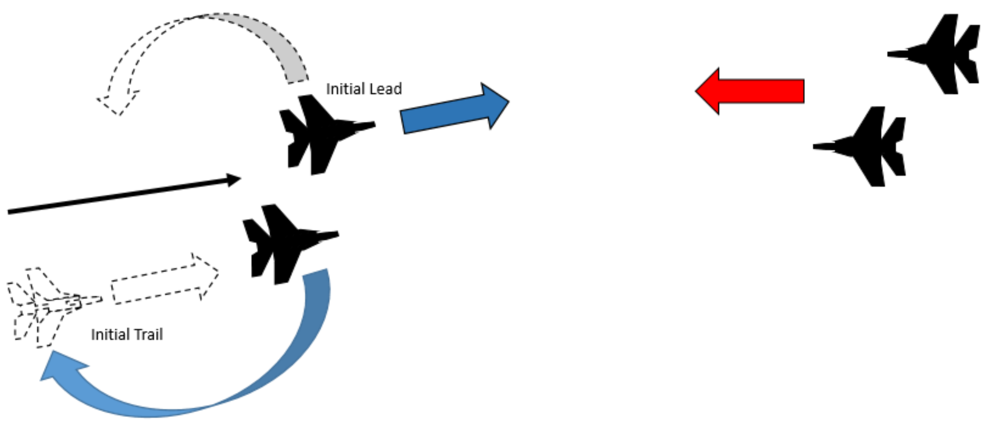



**Figure 17: Lead/Trail maneuver at BVR**

Finally, the pincer tactic, Figure 18 , splits the Blue flight. Each half of the flight maneuvers wide of the incoming enemy and comes into the approaching enemy flight at a maximum flanking angle (or Target Aspect Angle (TAA)) in order to attempt to get behind the enemy fighters. This tactic provides the best setup for follow-on into WVR engagement.

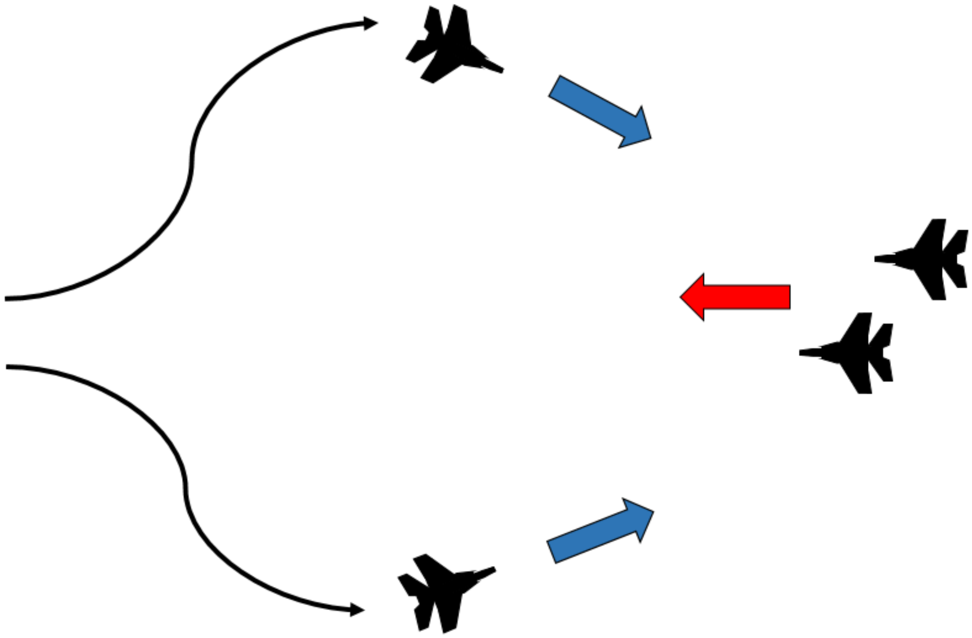


**Figure 18: Pincer tactic with a flight of two Blue aircraft at BVR**

For the purposes of this study, the focus of BVR tactics is on pincer and straight-in. WVR tactics are modeled using a simple decision engine that keeps the Blue fighter outside of the Red fighter's Weapon Employment Zone (WEZ), while attempting to maneuver the Blue fighter to a point that the Red fighter is within the Blue fighter's WEZ. Finally, BVR engagements typically begin 60 to 80 nautical miles from the targets and continue up to about 10 nautical miles from the targets, where the engagement then becomes classified as a WVR engagement (Houck, Whitaker, & Kendall, 1993).

Table 2 is a summary of the factors investigated in this air combat model and their operational ranges. Note that Blue Tactics is a categorical factor and the rest of the factors, though numeric, are discrete. This has implications for the analysis of the simulation outputs.

**Table 2: Summary of the study factors' operational ranges**

| Factors | Low | Central | High |
|---|---|---|---|
| Num. MRM (per Blue agent) | 0 | 4 | 8 |
| Num. SRM (per Blue agent) | 0 | 4 | 8 |
| Num. CUDAS (2:1) (per Blue Agent) | 0 | 8 | 16 |
| Num. CUDAS (3:1) (per Blue agent) | 0 | 12 | 24 |
| Num. Red fighters | 4 | 6 | 8 |
| Blue Tactics | Straight-In | N/A | Pincer |

Table 3 summarizes the main measures of effectiveness (MOEs) used in this study and provides the expected ranges of the responses.

**Table 3: Measures of Effectiveness (Responses) of the Simulation Calculation and Expected Value**

| | | | Expected Range | | |
|---|---|---|---|---|---|
| MOE | Name | AFSIM Output Metrics To Calculate MOE | Low | High | Remarks |
| MOE 1 | Time to Service Target Set | Average Simulation Time; PLATFORM_KILLED | 30 min | 120 min | Range is for the approx. 2 hour sweep scenario |
| MOE 2 | % Target Set Destroyed | Number Initial Red Targets; PLATFORM_KILLED | 80% | 100% | |
| MOE 3 | Weapon Effectiveness | Number Weapons Fired; Number Tgts Destroyed | 1 | 2 | Num Wpns Fired divided by Num Tgts Destroyed |
| MOE 4 | Standoff of Engagements | Avg. Engage Distance; | 20 nm (37 km) | 10 nm (18.5 km) | Expect that most air engagements occur at BVR |
| MOE 5 | Blue Vulnerability | Avg. Number of Hits by Red | 0 | 10 | The combination of tactics and weapons is expected to increase or decrease the vulnerability of a fighter in air combat. |
| MOE 6 | Engagement Results | Qualitative Engagement outcome | Engage WVR | Engage BVR | Expect that Blue attempts to stay at BVR |

**Data Collection Plan.**

The first step in analyzing the AFSIM output data is parsing the output files into quantified measures. For this purpose, we developed a post processor that works on the

comma delimited output files of AFSIM.  The AFSIM Post Processor, based on Excel with R script functions, provides a way to pull specific measures out of the data and collect them into a more usable data structure.  It also provides a summary data file that calculates averages, confidence intervals, medians, minimums, and maximums over all the simulation runs of one particular scenario or system.

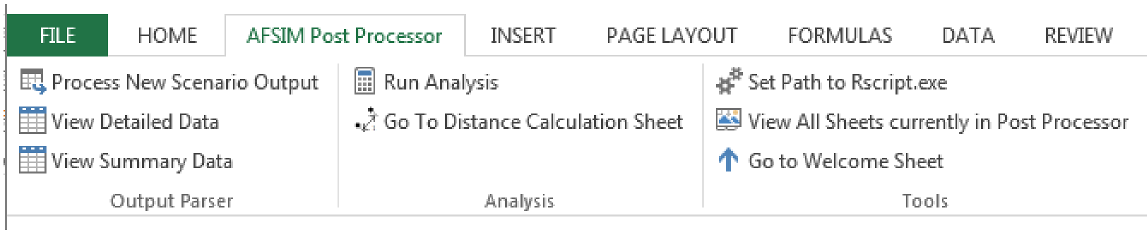



**Figure 19: AFSIM Post Processor Ribbon Options**

Before making production runs, AFSIM must be configured correctly to output the comma delimited files (file extension .csv) to a folder of the analyst's choosing.  We output all these comma delimited files to a separate folder under the same folder structure as the AFSIM scenario definition script files and name the output folder "output". Within AFSIM, the commands to output the .csv files are contained in the following AFSIM script block in the main script (or startup) file (Figure 20):

```
event_csvoutput
      relative_directory output/run%d/
      file_extension csv
end_event_csvoutput
```

**Figure 20: AFSIM Output Setup Script for Comma Delimited File Output**

The relative directory designates the path to the run output folders.  For the AFSIM Post Processor R script to work correctly, the "run%d" must be specified exactly as shown.  In the AFSIM script above, the %d tells AFSIM to create a folder named

"run[run number]" and then place the output .csv files from that particular run in them. If this is incorrectly specified, AFSIM will overwrite the files every time the simulation runs. The analyst can also specify events to output into the files. See the AFSIM wiki included in the AFSIM files for complete details on using the **event_output** block and how to enable output events. Figure 46, in Appendix A, is an example of one of the .csv output files, the WeaponTerminated_Data.csv file. There are actually 45 columns of data in this file, so we remove some of them for brevity. The AFSIM output contains many different .csv files and all are formatted differently.

After conducting the required runs of the different scenarios based on the analysis plan, the analyst must parse the AFSIM output files to extract the required information for analysis.

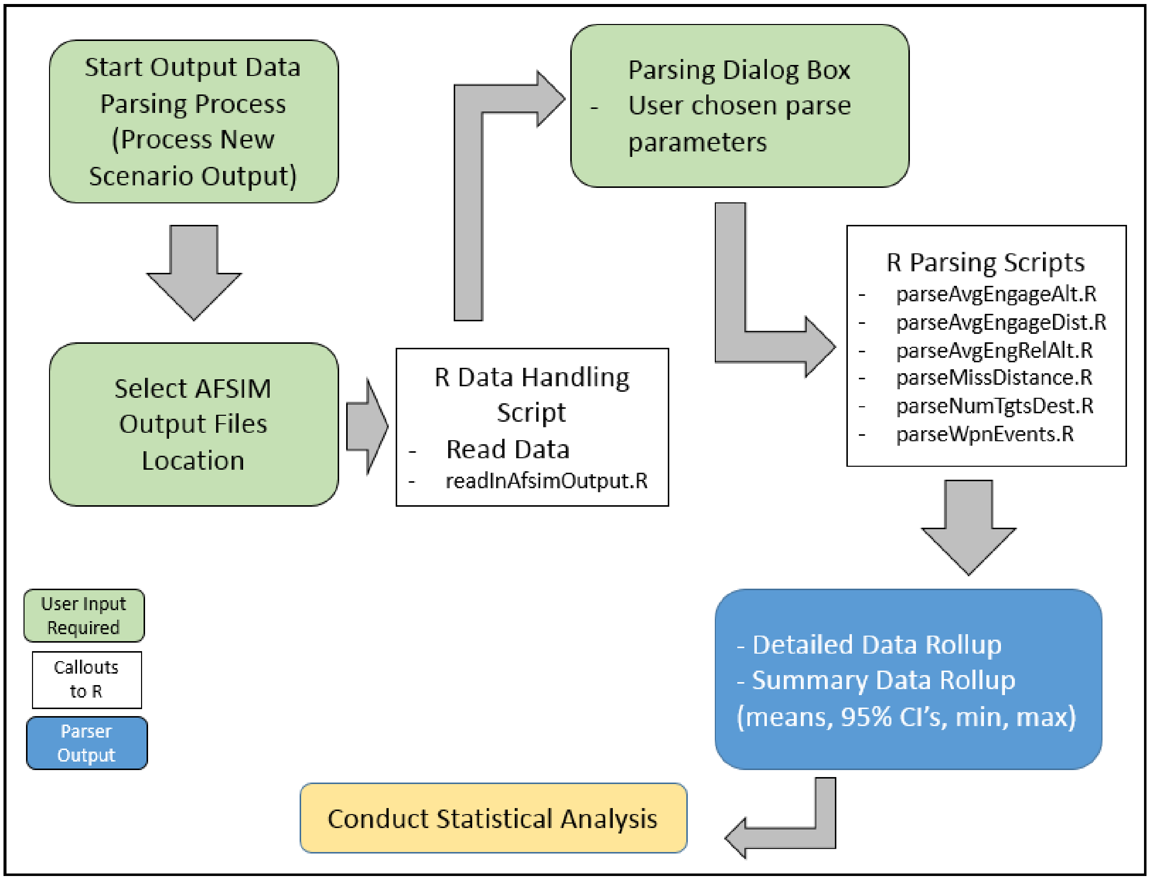



**Figure 21: AFSIM Post Processor output parser execution flow**

R scripts, called by the AFSIM post processor through Excel visual basic macros, conduct this post processing to parse the AFSIM output files (R Core Team, 2014). The flow of the program is presented in Figure 21. The post processor guides the analyst through selecting the location of the AFSIM .csv output files. To begin, the analyst just clicks on the "Process New Scenario Output" button in the Excel Ribbon (as shown in Figure 19). Then the open dialogue box appears and the analyst can select the location of the output files, as shown in Figure 22.

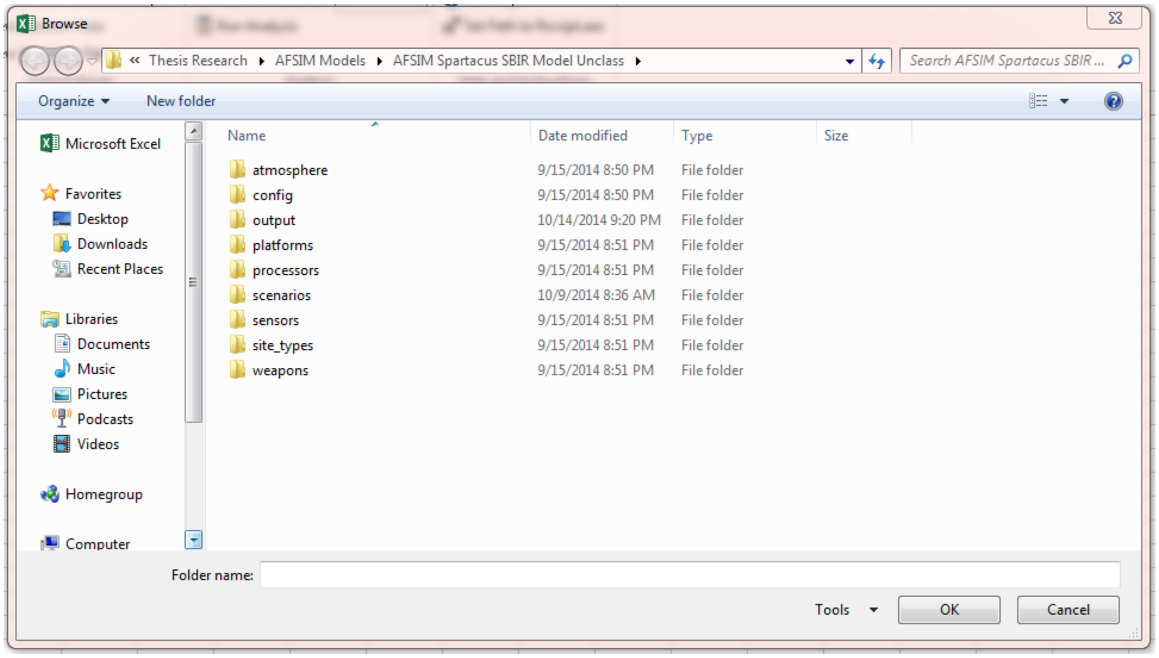



**Figure 22: Open Folder dialog that allows selection of the AFSIM output file location**

The AFSIM Post Processor works by reading each of these .csv files into the R environment in the form of a data frame, which is a special data structure in R formatted much like a matrix, but has special attributes and methods useful for data manipulation. Once the files have been put into the correct format for R, the post processor displays a dialogue box that allows selection of the responses, platforms, weapon types, and target types to parse the output data by as shown in Figure 23.

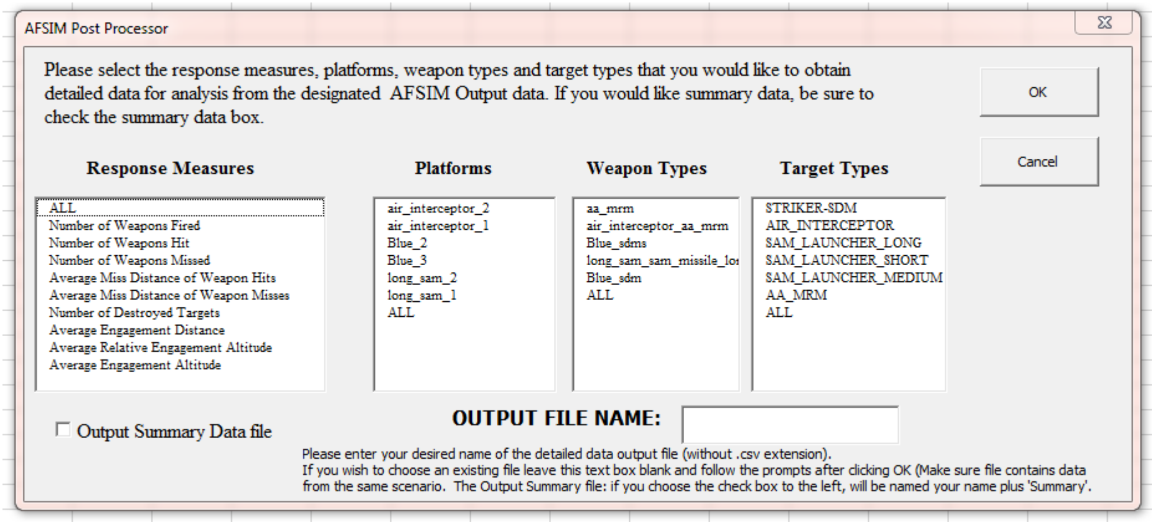



**Figure 23: Parse data selection - All platform, weapon type, and target type names are extracted directly from data; Response Measures are pre-programmed.**

Currently the post processor is limited to just these three of the AFSIM comma-delimited output file types (WeaponTerminated_Data, WeaponFired_Data, and Platform_Data), but can easily be expanded to read in any of the files containing response data of interest from AFSIM.

The post processing script then uses the analyst's choices for responses and parses the data for all the combinations of the chosen platform, weapon, and target type of interest, with the resulting detailed data frame formatted as in Figure 24 below:

| Firing.Platform | Wpn.Type | Target | Stat.Name | Run | Value | Half.Width | Variance | Median |
|---|---|---|---|---|---|---|---|---|
| air_interceptor_3_ | aa_mrm | STRIKER-LWS | WEAPON_FIRED | 1 | 6 | 0 | 0 | 0 |
| air_interceptor_3_ | aa_mrm | ALL | WEAPON_FIRED | 1 | 6 | 0 | 0 | 0 |
| air_interceptor_3_ | ALL | ALL | WEAPON_FIRED | 1 | 6 | 0 | 0 | 0 |
| Blue_2 | hel | AA_MRM | WEAPON_FIRED | 1 | 9 | 0 | 0 | 0 |

**Figure 24: Detailed Data Frame Created by the AFSIM R Post Processing Script**

Where the **Firing.Platform** is the platform of interest, the **Wpn.Type** is the weapon type fired by the platform, Target is the platform's target for that engagement,

and **Stat.Name** is the response variable of interest. The post processor calculates most of the measures by simply sub-setting the appropriate data frame and counting the number of observations, using different variations of the following R command. Note that the + symbols at the start of a new line just indicate to R that the new line is actually a part of the original line; in other words the entire command in Figure 25 may be written on one line with the + symbols excluded:

```
>  detailedData$Value <- nrow(subset(frameName, subset =
+ (platform  == p & wpnType == w & Target.Type == t & run
+ == i & event == e)))
```

**Figure 25: R "nrow" Function Example Syntax for Counting Event Occurrences of a Certain Subset**

The “nrow” function used in conjunction with the “subset” function does most of the work. The **Half.Width** column is a 95% confidence interval half-width for the particular data measure. Not all measures will yield a confidence interval, because they do not have ‘within replication’ variance.

Once the program creates the detailed data frame from the analyst’s choices, the post processor script saves it to a location and name of the analyst’s choosing.

To calculate summary data across replications, the post processor uses a function to summarize the data contained in the detailed data frame. The output from the summary function shows the roll up of each of the measures across all the runs of the simulation conducted. The output “summaryData” data frame is formatted the same as the “detailedData” data frame but includes the columns min and max. An example of the format is shown in Figure 26.

| Firing.Platform | Wpn.Type | Target | Stat.Name | Average | Half.Width | Variance | Median | Max | Min |
|---|---|---|---|---|---|---|---|---|---|
| Blue_2 | sdm | AIR_INTERCEPTOR | WEAPON_FIRED | 5 | 0.79806751 | 12.86075949 | 4.5 | 10 | 0 |
| Blue_2 | sdm | AIR_INTERCEPTOR | WEAPON_HIT | 2.825 | 0.45863944 | 4.247468354 | 3 | 7 | 0 |
| Blue_2 | sdm | AIR_INTERCEPTOR | WEAPON_MISSED | 1.4666667 | 0.30255767 | 1.371751412 | 1 | 5 | 0 |
| Blue_2 | sdm | AIR_INTERCEPTOR | WEAPON_TERMINATED | 5 | 0.79806751 | 12.86075949 | 4.5 | 10 | 0 |
| Blue_2 | sdm | AIR_INTERCEPTOR | Avg Engage Distance | 35667.464 | 2486.97068 | 28237281.63 | 33577.97 | 49023.91 | 29822.75 |
| Blue_2 | sdm | AIR_INTERCEPTOR | Avg_Engage_Altitude | 12871.665 | 100.61777 | 46220.1052 | 12921.6 | 13138.39 | 12289.14 |
| Blue_2 | sdm | AIR_INTERCEPTOR | Avg_Engage_Relative_Altitude | -598.9508 | 294.511295 | 395991.1374 | -180.629 | -152.266 | -2101.7 |
| Blue_2 | sdm | AIR_INTERCEPTOR | Number Of Targets Destroyed | 2.625 | 0.44560606 | 4.009493671 | 3 | 7 | 0 |

**Figure 26: Summary Output Data data-frame format**

Currently the post processor only parses the AFSIM output for the factors: Platform, Weapon Type, and Target Type. In addition, the post processor only calculates the responses: WEAPON_FIRED, WEAPON_HIT, WEAPON_MISSED, WEAPON_TERMINATED, Average Engagement Distance, Average Engagement Altitude, Average Relative Engagement Altitude, Number of Targets Destroyed, Simulation Time that Platform is destroyed (if destroyed), and Average Miss Distance of Weapons (Hits and Misses).

Other measures, such as First Detect, are easily added if the analyst specifies the associated events for output in the .csv output files from AFSIM. Some measures must be parsed out of the flat text files (such as Behavior Tree data), rather than the comma-delimited files, because AFSIM simply does not yet output those events to the .csv files (see the AFSIM wiki page on event_output). However, AFSIM Version 1.8 does have an included Behavior Tree tool, called GRIT (Graphical RIPR Interface Tool), which provides a tree visualization and a time slider to show which behaviors are active at what simulation times for a particular run of the simulation scenario (shown in Figure 27).

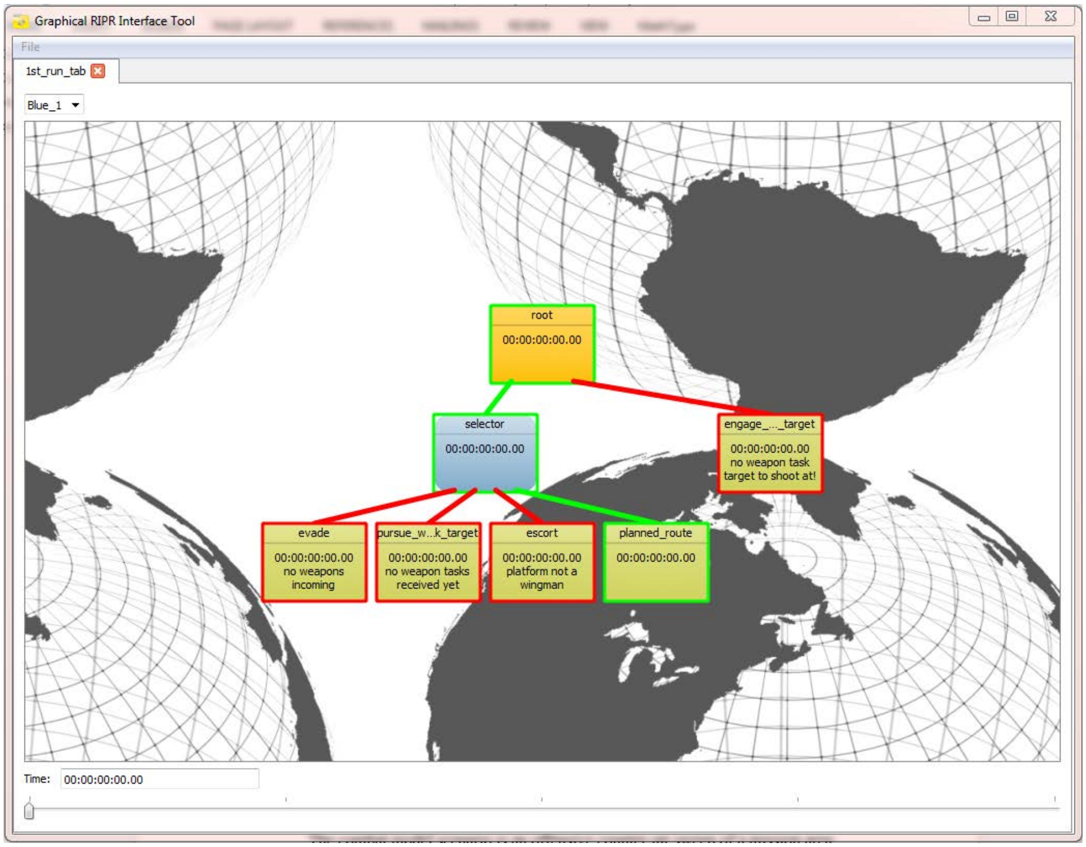



**Figure 27: AFSIM 1.8 Behavior Tree Visualization Tool, Graphical RIPR Interface Tool (GRIT)**

## Simulation Scenario

The combat model scenario is an offensive counter-air sweep of a mission area, which takes place in a theoretical country. Much of this scenario definition comes from the previous work of Charles River Analytics for the "Situational Awareness Model for Pilot-in-the-Loop Evaluation" with some minor variations (Mulgund, Harper, Guarino, & Zacharias, 1999). Specifically, we adjust the number of agents on both the Blue and Red sides for the purposes of this study.

The general scenario description is one flight of two Blue fighters have the mission to clear an area by moving to contact with enemy fighters in order to support a

follow-on strike on a deep target. The strike mission objective can be a highway bridge or factory or some other type of high value target. Along the route, the fighters may encounter enemy air threats or enemy air defense threats and will work to eliminate these threats so that follow-on bombers can destroy the objective.

**AFSIM Scenario Implementation.**

The simulation is run in the AFSIM framework, a mission-level combat modeling simulation framework. AFSIM scripting files use AFSIM commands to control the simulation execution, call specific models, and define the platforms and the environment in which the platforms exist. For this simulation scenario, the files are categorized into different overall categories and stored within folders of that category. Figure 28 depicts a snapshot of the project browser pane of the AFSIM integrated development environment (IDE). The IDE is the tool that is used to interact with the various code files.

Much of the AFSIM scenario files are adapted from the work of the Air Force Research Laboratory's Sweep Mission Scenario for the Spartacus Study (Geaslen & Panson, 2014).

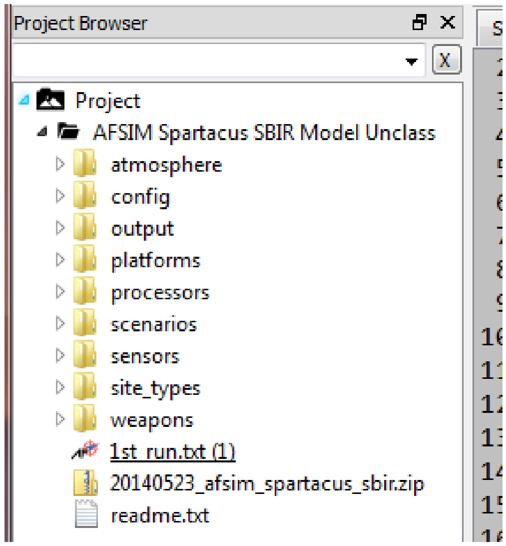



**Figure 28: AFSIM Simulation Scenario Scripting Folder Structure**

The main simulation control file that starts up the simulation and controls the execution of the simulation is shown in Figure 28 as "1st_run.txt". This file tells AFSIM where to find the main scenario file, the setup file, and the event output file. The file also carries commands on the number of runs, location to put the output files, random number generation, and variable definition. The main scenario file is located in the "scenarios" folder and contains the commands to instantiate specific instances of platforms, their locations, and routes. The "config" folder contains the setup file, the event output file and the terrain file. The setup file gives AFSIM the paths to critical files as well as commands to "include" all the script files necessary to run the simulation. The event output file tells AFSIM which events to record in the output files. The terrain script file includes references to any terrain files, such as Digital Terrain Elevation Data (DTED) files, used in the simulation. For our scenario, we chose to use no terrain to help simplify factor interactions. Figure 29 is an example of the AFSIM commands included in the "1st_run.txt" startup file.

```
define_path_variable CASE 1st_run
log_file output/$(CASE).log
# setup file includes the platforms
include config/setup.txt
include config/event_output.txt
# Select one of these scenarios to run
#include scenarios/Scenario-Striker-LWS.txt
#include scenarios/Scenario-Striker-SDM.txt
include scenarios/Scenario-Striker-Base.txt

event_output file output/$(CASE).evt end_event_output
dis_interface
	record output/$(CASE)%d.rep
	mover_update_timer 5.0 s
	suppress_directed_energy_data on
end_dis_interface

event_csvoutput
	relative_directory output/run%d/
	file_extension csv
end_event_csvoutput

final_run_number 5
random_seed 234546 #used for one iteration
generate_random_seeds 367 #used to generate a stream of random seeds for each run using the same base seed up to the final_run_number
end_time 120 minutes
```

**Figure 29: Example Startup File for the AFSIM Simulation Control**

A # character indicates comments. Script blocks always include an “end” statement. AFSIM commands are bold and usually followed by a setting in light text. The AFSIM scripting language also includes all the basic programming control structures such as for and while loops and if-then statements as well as model specific commands.

The rest of the file structure includes all the class files that define classes of platforms, weapons, sensors, and weapons. Because AFSIM is object oriented in nature, each of the script files is, in essence, a definition of a class of objects, which are then instantiated in other class definitions or in the simulation scenario file as specific instances of that object. This gives great flexibility to the simulation developer in that a platform, or any other type of object, needs only be coded once in a class definition. Then multiple instances of that platform, say multiple Blue fighters, can be specified in the scenario file or in another class definition. Specific instances of an object can have attributes set specific to that instance or even have attributes added that do not exist in the base class of that object.

In Figure 28, the “site_types” folder contains all the “intelligent” agent processor class definitions used for Blue agent decision making. The “decision-making” processors for the Red agents are contained in the “processors” folder. The rest of the folders are self-explanatory.

Figure 30 shows a screen shot from the Visual Environment for Scenario Preparation and Analysis (VESPA) playback visualization utility that shows a particular run of the simulation graphically. This playback utility, included in the base AFSIM distribution, is useful in conducting verification and validation as will be discussed later.

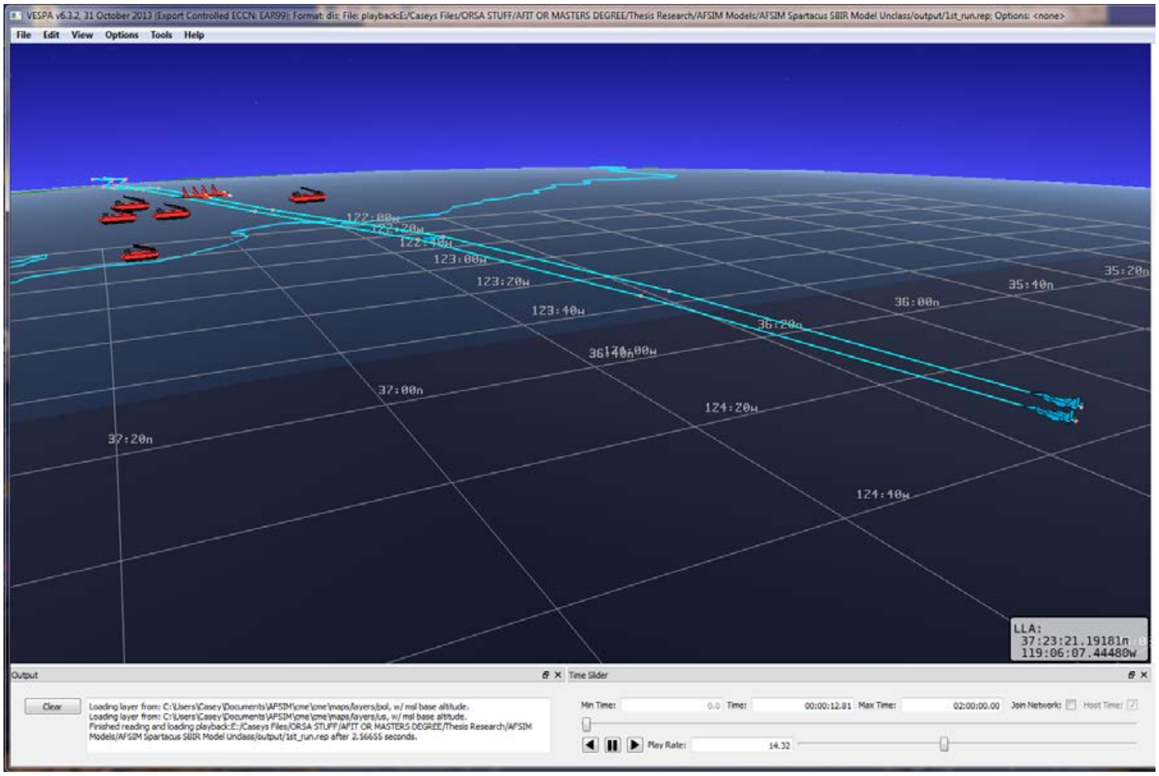



**Figure 30: VESPA Playback Snapshot of Simulation Run of AFSIM scenario**

The VESPA utility is also useful in development of a scenario as it allows graphical placement of platforms and drawing of routes. As is shown, Figure 30 shows the routes programmed for the Blue fighters to take. Because this is a playback of an actual simulation run, there is scripting code included in the files that draws line objects to show the routes as the respective agents fly. These lines disappear as the Blue agents react to the various Red platforms and then reappear as the Blue agents move back onto the planned routes making it easy to tell when the agents use the “planned-route” behavior. The view within VESPA can be zoomed, panned and rotated to view different actions from different angles.

**Blue Agent Decision Making Behavior.**

The fighters have several planned courses of action (i.e., primary, secondary, tertiary) for ingress, sweep, and egress from the mission objective area. Throughout the mission, each agent goes through several different states. Each agent can be in a search state, looking for possible threats and targets. Search happens on both ingress and egress. The agent moves to a detection state upon target identification. The engagement state is when the agent selects a weapon and fires at the detected target. Finally, each agent conducts a defensive reaction if the agent detects an incoming threat.

AFSIM includes a pilot mental model, the Reactive Integrated Planning aRchitecture (RIPR), that uses job boards and behavior trees to provide an artificial intelligence framework for creating flexible agent behaviors in reaction to the simulation environment (Zeh & Birkmire, 2014). Job boards are essentially tasking algorithms that allow a commander agent to allocate different sensor tracks as "tasks" to subordinates for engagement. The algorithm performs a weapon-target assignment and passes a task to the designated subordinates. The "quantum-tasker processor" is the object that implements this task creation (generation), evaluation and allocation. Figure 31 depicts the flow of the agent decision making from the sensing of a possible track to assignment of the task to a Blue agent and then evaluation by the Blue agent's Behavior Tree. The Behavior Tree is discussed later in this section.

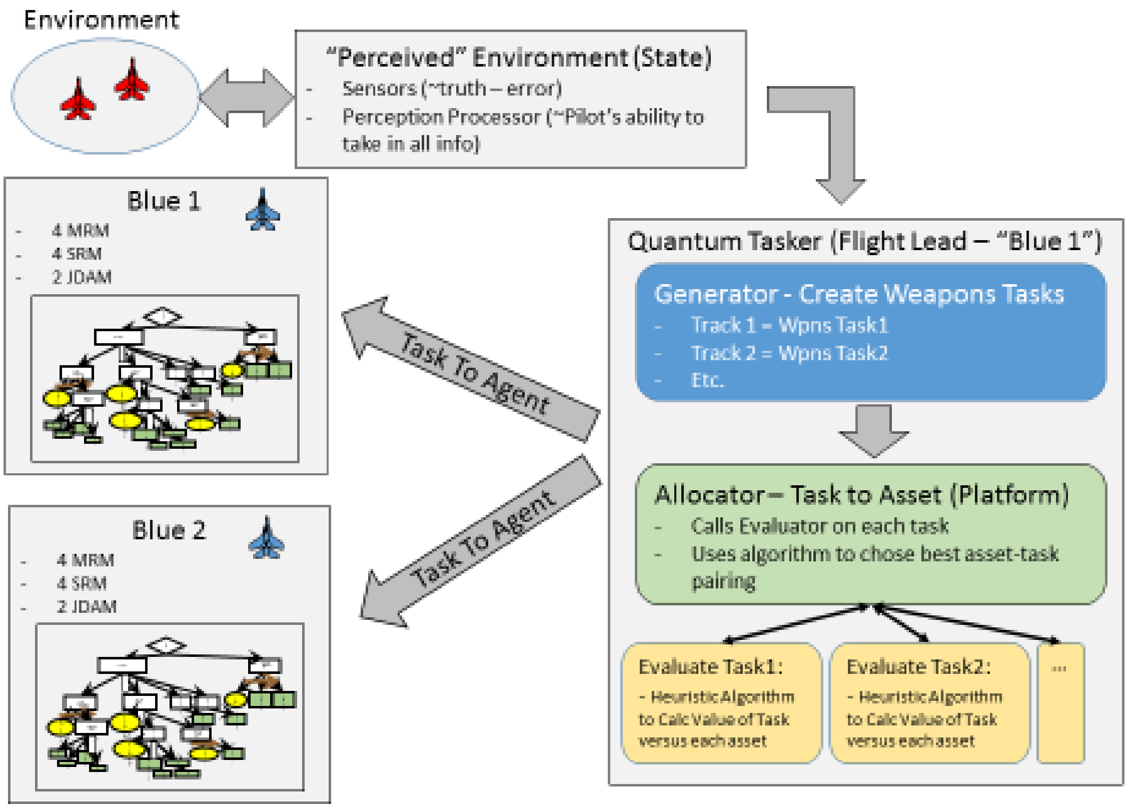



**Figure 31: Blue Agent Decision Making Architecture within the RIPR model in AFSIM**

For this scenario, the quantum-tasker processor implemented within the Flight Lead agent creates simple weapons tasks from every track perceived by the Blue agents' sensors. The Evaluator script then takes each task and evaluates it against each subordinate asset. The quantum-tasker processor is set to define assets as subordinate agents to the Flight Lead. Finally, the Allocator uses the value developed for each asset-task pair to find the optimal allocation of tasks and then assign those tasks to the assets.

Each weapons task is evaluated per asset by finding the maximum value weapon carried on that asset against that task. The pseudo-code is shown in Figure 32. The

heuristic considers the closeness of the target to the asset, the probability of hit/kill for the asset's weapon versus the specific target, and the effective range of each weapon on the asset.

```
Initialize all variables;
Value = 0;
RangeToTarget = SlantRangeTo(EnemyTrack);
WVRdistance = 10 Nautical Miles;
For each weapon on asset i
  Value = Value + (1/(RangeToTarget/WVRdistance));
  Value = Value + WeaponPSSK;
        #PSSK = Probability Single Shot Kill
  If Target is within range of weapon
        Value = Value + 1;
  Endif
  If Target is WVRdistance and weapon is a short range
                                                              weapon
        Value = Value + 1;
  Endif
  If Target is an aircraft and is outside of the mission
                                                                area
        Value = 0;  # Target is removed as a feasible target
  Endif
End For Loop
```

**Figure 32: Pseudo-Code for Evaluator heuristic rules**

The allocator within the quantum tasker processor runs once every ten simulation seconds. Each time the allocator runs, it calls the evaluator for every asset (agent) – task (target) pair. There are several allocator algorithms pre-built into the RIPR quantum model. The scenario implemented in this study uses a custom algorithm to provide the asset-task assignment. The allocation algorithm fully enumerates the possible solutions to the assignment of asset to tasks and then picks the largest valued combination of the

assignments. The algorithm also prevents assets from executing the same targets. The pseudo-code is shown in Figure 33.

```
Initialize variables – TempTotalValue = 0, MaxTotalValue = 0;
For each asset_task_value I  #iterate over every combination of tasks
  For each asset_task_value j
    If i not equal to j #Disallow assets assigned same task
        TempTotalValue = asset_task_value i +
                                      asset_task_value j;
        If TempTotalValue > MaxTotalValue
             MaxTotalValue = TempTotalValue; #Find the
                                              largest value
             If the value of any of the tasks in the
                                    asset_task combination = 0
                 The target has no value, cancel the task;
             Endif
         Endif
     Endif
   End for loop
End for loop
```

**Figure 33: Allocator Custom Full Enumeration Pseudo-code**

As discussed in Chapter 2, Behavior Trees are a rule-based set of reactive behaviors built in a tree form that allow flexible entity behaviors. The Behavior Tree developed for this scenario is depicted in Figure 34.

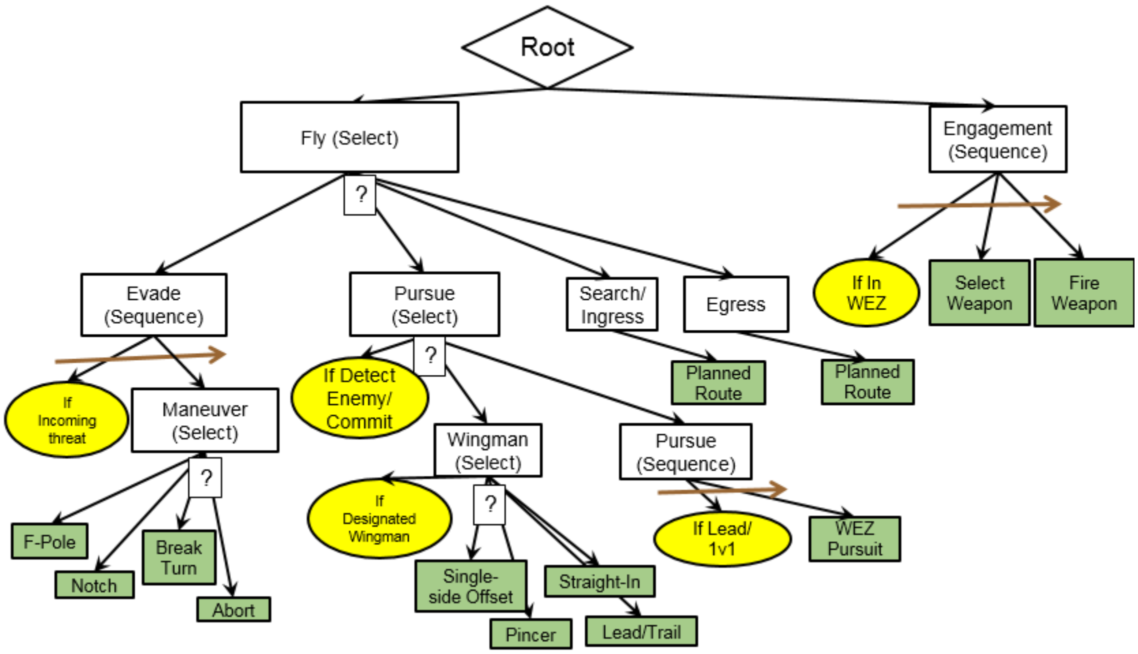



**Figure 34: Agent Behavior Tree for the Sweep Mission Scenario**

On every update, or scheduled evaluation of the tree, the root node fires, attempting to execute all of its child behaviors. The "Fly" node, being a select node, attempts to select one of its children for running. The node starts on the left and stops as soon as it reaches a subordinate node that returns running or true. In this way, the "Evade" sub-behavior gets a priority evaluation. Inside the "Evade" sequence node, the node attempts to run the subordinate nodes from left to right in sequence. If a subordinate returns false, the "Evade" node returns false and cannot run. If the situation meets the "Incoming Threat" pre-condition, then the node moves to the "Maneuver" select node, which attempts to select one of the four basic behaviors available to it. The basic behaviors are simply scripts written in AFSIM's scripting language (a high-level language similar to C++) that execute some action. If the action is impossible to execute, given the current state of the environment and the fighter, that action will return false.

The “Maneuver” select node then moves on to the next basic behavior to attempt that action and continues until it reaches a behavior that executes or returns false as unable to execute any subordinate behaviors.

In this manner, the “Fly” node tries each of its subordinates. Note that the Ingress and Egress Nodes, depending on the location of the fighter and the overall state of the simulation, will always be able to fire. These nodes become the “default” for the “Fly” node.

Simultaneously, the root node is trying to run the “Engagement” sub-behavior. This behavior fires the weapon assigned to the current target. If the weapon cannot be fired (the target is outside the Weapon Employment Zone (WEZ)) of the weapon or the target’s flight characteristics make the track quality or probability of kill for the weapon too low), the node returns false and the weapon does not fire.

**Red Force Composition and Behavior Engine.**

The Red side within the simulation consists of fighters and air defense vehicles carrying surface-to-air missiles on the ground. Red aircraft each carry the same fire-control sensor and have the same fourth-generation fighter flight characteristics as the Blue aircraft. Weapons compliment consists of four medium range missiles and six short-range missiles, which is consistent with current defensive counter-air (DCA) mission load outs. The missiles themselves are the same type as carried by Blue in the baseline weapon configuration. The number of Red fighters varies within the range of two to six total fighters flying in pairs throughout the sweep mission area. After initial analysis, we adjust the range of number of Red fighters from four to eight, as we discuss in more detail in chapter 4. There are three types of Red air defense systems within the

scenarios: long-range, short-range, and medium-range SAMs. Each is kept at a fixed level across all the scenarios: one Long-range, two short-range, and one medium-range SAM. The Red decision-making engine driving Red behaviors uses the AFSIM Finite State Machine model for entity behaviors. Figure 35 shows the four possible states that each Red agent can be in at any time. The baseline behavior is to conduct pre-programmed Patrol/Search behavior, such as follow a route. Once a radar track is established, the agent moves to the detected state. In the detected state, the agent maneuvers towards the target in order to make the target engage-able.

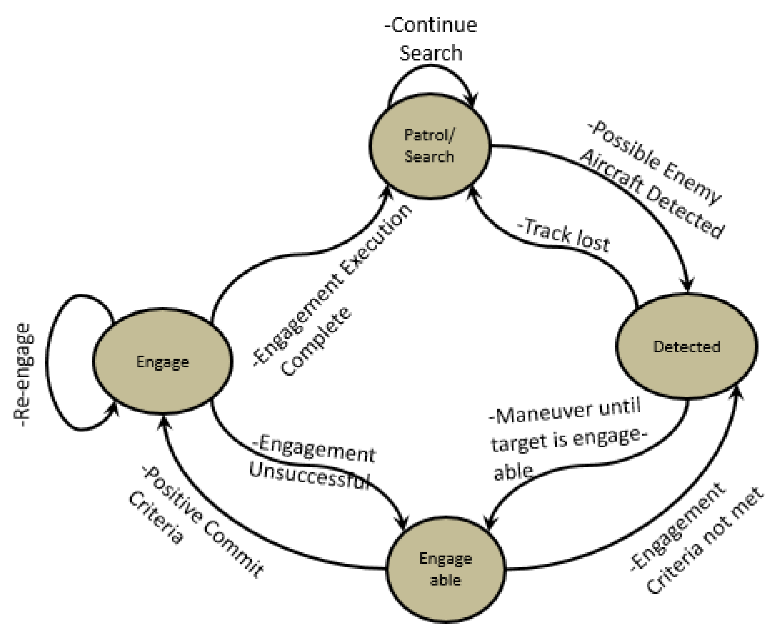



**Figure 35: Finite State Machine Diagram for Red Agents (Four Possible States) Scenario Summary**

Once the track is determined to be an enemy track, the agent moves to the engage-able state in which the script checks if the track meets the engagement criteria (position, altitude, speed, etc.), or in other words whether the track is in the engagement zone of the

Red agent's weapon.  If engage-able, the agent moves into the Engage state and attempts to fire its weapon at the target track.  At any time, the logic in the state machine allows the agent to move back and forth between the states according to the diagram in Figure 35.

**Scenario Summary.**

This scenario offers several advantages for analyzing different weapon systems. One advantage is that previous research provides a detailed definition of this scenario (Mulgund, Harper, Guarino, & Zacharias, 1999).  Another is that it offers a framework within which to test several very different kinds of agents against each other using the same behavioral architecture for each agent or different behavioral architectures for each agent.  Finally, different weapons systems and platforms can be added and subtracted quickly, creating different scenarios within which to show agent actions under the different behavioral architectures.

Data on tactical behavior is gathered by using the GRIT visualization interface to gather information on which behaviors are used and how often.  This information details how implementing the new weapon within the simulated air combat scenario influences the agent choice of behavior.

## Verification and Validation

We conducted verification visually using the VESPA playback utility as well as analytically using a check of the output data.  This is a time consuming process and requires sampling from individual scenarios.  Visually, the analyst must check that the agents execute the tactics correctly and in the correct context.  Checking the data, several

questions are asked: “Are all data factors being populated?”, “Are the output numbers realistic given the scenario?” and “Are there any questionable output numbers?”. Many times errors can be found quickly, but finding the underlying causes requires a systematic trial and error approach to correcting the simulation code. Verification of the response surface model achieved with the designed experiment is an additional technique. The analyst randomly conducts confirmatory runs on predicted values at several different settings of the factors. If the responses fall within a statistical prediction interval of the predicted value, the model is performing satisfactorily.

We complete validation of the simulation model in this study chiefly through subject matter expert (SME) analysis. We conduct our simulation runs and then have SMEs from AFRL/RQ and Lockheed examine key outputs, including some of the visualizations and provide feedback on agent behaviors and weapons performance.

As of the writing of this paper, HAF/A9 and AFRL are conducting a more in-depth validation for the underlying combat models within AFSIM. HAF/A9 is evaluating AFSIM for possible inclusion of the framework into the Air Force Standard Analysis Tool Kit (AFSAT).

**Analysis Plan**

The analysis of the weapon system using this simulation scenario is undertaken using a designed experiment approach, described in more detail in Chapter 4. The main steps of the execution of the analysis follow:

1. Develop and run separate simulation scenarios for a specified number of iterations for each design point, or treatment combination, in the designed experiment matrix. This step generates the data for analysis. We extract the data using the AFSIM post processor.

2. Conduct initial analysis using qualitative visualization analysis of each separate simulation scenario. The initial analysis yields insights about tactics used, efficiency, and effectiveness of the different weapon systems involved.

3. Conduct ANOVA and construct a statistical model using the designed experiment treatment combinations. The model yields statistically valid and useful insights about the contribution of each factor (number of weapons by type, number of Red agents, and tactics used) as well as the significance.

4. Summarize and report the results.

### Summary

The overall methodology of this study is to use an agent-based simulation, AFSIM, to investigate the effects of using a new air-to-air missile in air combat. We have described the tactics used in BVR air engagements and developed specific metrics for use in analyzing the weapons system. The scenario used in this study is specifically a sweep mission, but there are many other scenarios that could be investigated, for instance, the Defensive Counter Air, Suppression of Enemy Air Defense, Deep Interdiction, and others. The sweep mission is chosen because it is well defined, accessible, and has all the elements of both air-to-air and air-to-ground combat that allow sufficient exploration of the main factors of interest.

We provide many of the technical details of AFSIM here in order to show by what method intelligent agents may be configured to simulate air combat. A somewhat simplistic weapon-target assignment algorithm is used for the higher-level cognitive functions of a flight leader. Algorithms and heuristics exist that are more efficient and an approximate solution heuristic may even be preferable, given that we are modeling human decision-making, but are beyond the scope of this thesis. A reactive, rule-based

behavioral framework, Behavior Trees, are used to model Blue agent actions within the air combat environment, while Red agent actions are governed by a finite state machine. The BTs allow actions that are more complex and a certain degree of cooperation between the Blue agents that provides more realistic agent decision-making behavior.

Finally, the heart of the analysis is the designed experiment approach. Initial insights are gained by a "quick-turn" review of the resulting run data using both visualization and statistical techniques, but the statistical model constructed from the experimental treatment combinations data provides additional insights pertaining to the relevance and significance of the experiment factors to the complex system of air combat.

# IV. Analysis

## Overview

This chapter details the analysis results of our evaluation of the data obtained from a designed experiment using the simulation scenario described previously. The logic behind the experiment design is to provide insight into which factors are producing significant effects and a general idea of the direction of those effects, as well as how the factors interact with each other, to produce the observed effects in the air combat scenario. The designed experiment (DOE) approach provides a clearer picture of the effects of mixing weapons and the extent of involvement that different tactics have within the combat scenario. Ultimately, this analysis is designed to give insights that help answer the initial questions we start with in Chapter 1. The DOE approach we use here is a screening design. Ideally, this initial experimentation is used as a basis for further simulation experimentation based on the insights gained.

## Designed Experiment Analysis.

We consider several experimental designs due to the unique nature of the factors involved in this analysis. The carrying capacity of the aircraft on which the weapons are carried limits the first three factors, number of each type of air-to-air missile used. The interaction between each of the other missiles carried also limits the number of a specific missile carried, as well. For example, if eight MRMs are carried on an F-15, then no other air-to-air missiles can be carried. Furthermore, these factors are discrete in the sense that a fractional missile is not logical. These three factors represent a mixture. The

number of Red aircraft is also discrete. We choose to limit the choice of tactic used to a categorical variable; the tactic is either pincer or straight-in, in order to show the effects of completely distinguishable tactical courses of action. This set of factor characteristics makes the choice of experimental design somewhat complicated.

Two designs are considered: a mixture design of the three missile factors combined with a full $2^2$ factorial screening design and a computer generated D-optimal design with imposed constraints that treats the missile factors as non-mixture factors and prohibits infeasible mixtures three missiles. We discuss each design briefly, but choose to use the computer generated D-optimal design, which does not use a mixture design.

The first design, a mixture design in conjunction with a full factorial screening design involves conducting a mixture simplex experiment for the three missile factors at each of the points of the factorial screening experiment, where each factorial point represents a different combination of the factor levels of the two non-mixture factors, number of Red aircraft and tactic used. This design has the advantage of being the most transparent in terms of comprehension due to the design construction method. This design also has a decent variance structure. However, it has several disadvantages. First, the mixture design/full-factorial combination requires a large number of design points. (Approximately seven design points for each mixture multiplied by six points for each point in the $2^2$ factorial plus two center runs is 42 total design points.) Each design point in AFSIM must be a separately programmed and run scenario because AFSIM does not have an experimental engine to allow changing variables between simulation runs during runtime. Additionally, this design has a troublesome aliasing structure, meaning that some effects are confounded within the design. Finally, the mixture design requires

continuous factors. The mixture design can be adjusted to accommodate our discrete missile factors, but this results in a non-optimal variance structure.

The next design, which is the design used in this analysis, is a computer generated D-optimal design with designated “disallowed” points. The disallowed points are the infeasible combinations of the missile factors (i.e., 8 MRM with 8 SRM, etc.). Computer software is used to generate the design and conduct analysis on the outputs, specifically JMP 10.0 (SAS Institute Inc., 1989-2007). The computer software uses a constrained optimization technique to generate a design with the least amount of design points required for the designated factors with an optimal variance structure. In this case, we use the D-optimal criteria, which seek to minimize the variance of the model parameter estimates within the design region. Although the variance structure of this design is not as desirable as the mixture/factorial design, this design has fewer numbers of points and a better aliasing structure, meaning that effects are more readily apparent because they are only partially aliased with other effects, as compared with the mixture/factorial design. The final design matrix used for this analysis is in Table 4. One last note, the design in Table 4 is a single replicate. For this analysis, we conduct twenty replicates of each of the eighteen design points to provide a solid measure of error.

**Table 4: JMP Generated Custom D-Optimal Screening Design**

| Run | Number of MRM | Number of SRM | Number of CUDAS | Number of Red Fighters | Blue Tactic Used |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 4 | Pincer |
| 2 | 4 | 0 | 0 | 4 | Pincer |
| 3 | 2 | 2 | 0 | 4 | Pincer |
| 4 | 0 | 4 | 0 | 6 | Pincer |
| 5 | 2 | 0 | 4 | 6 | Pincer |
| 6 | 0 | 2 | 4 | 6 | Pincer |
| 7 | 4 | 0 | 0 | 8 | Pincer |
| 8 | 0 | 4 | 0 | 8 | Pincer |
| 9 | 0 | 0 | 8 | 8 | Pincer |
| 10 | 0 | 4 | 0 | 4 | Straight-In |
| 11 | 0 | 2 | 4 | 4 | Straight-In |
| 12 | 0 | 0 | 8 | 4 | Straight-In |
| 13 | 0 | 0 | 0 | 6 | Straight-In |
| 14 | 4 | 0 | 0 | 6 | Straight-In |
| 15 | 0 | 0 | 8 | 6 | Straight-In |
| 16 | 0 | 0 | 0 | 8 | Straight-In |
| 17 | 2 | 2 | 0 | 8 | Straight-In |
| 18 | 2 | 0 | 4 | 8 | Straight-In |

**Simulation Scenario Analysis Results.**

The above experimental design is applied using the scenario described in Chapter 3. From the simulation output data, we construct a statistical model for each MOE that allows us to look at the significance of the effects of each of the factors as well as determine some mean values with 95% prediction intervals.

In its current configuration, there are two possibilities for the ratio of CUDA to MRM/SRM that can be replaced on the standard weapons rails of fourth generation aircraft. Each weapon rail can carry either two CUDA missiles or three CUDA missiles as opposed to just one MRM or SRM. In order to build statistical models that accurately reflect the changes in the MOEs due to the changes in the levels of the number of each

type of missile carried, it is necessary to adjust the scale of number of total missiles available in the scenario versus the number of Red aircraft targets in the scenario. In an effort to keep this scenario as simple as possible, we limited the number of Red aircraft to a maximum of eight. Although fourth generation aircraft may carry up to eight MRM/SRM, or sixteen CUDA in a two to one replacement scheme, this provides an overmatch in capability to number of targets. In other words, it is difficult to assess if changing the number of CUDA from sixteen per Blue aircraft to eight per Blue aircraft is having any effect. At a maximum, there are only eight targets. The Blue side may have up to thirty-two total missiles (sixteen missiles on two Blue aircraft). Our solution was to decrease the number of missiles available, but maintain the ratios. We limited the number of MRMs or SRMs available to four per Blue aircraft, as shown in Table 5. This means a maximum of eight CUDA for a two to one replacement scheme.

**Table 5: Summary of the study factors' re-scaled operational ranges**

| Factors | Low | Central | High |
|---|---|---|---|
| Num. MRM (per Blue agent) | 0 | 2 | 4 |
| Num. SRM (per Blue agent) | 0 | 2 | 4 |
| Num. CUDAS (2:1) (per Blue Agent) | 0 | 4 | 8 |
| Num. Red fighters | 4 | 6 | 8 |
| Blue Tactics | Straight-In | N/A | Pincer |

Assessing a three to one replacement scheme requires an addition of many more Red aircraft, which is beyond the scope of our stated simple analysis scenario. This in itself shows a potential benefit of having a smaller, lighter missile: increased cargo capacity. This benefit is discussed further in a later section.

## CUDA 2 to 1 Replacement Scenarios.

### MOE1 Time to Service Target Set.

As discussed in Chapter 1, MOE 1 is measured at the time that both Blue agents arrive at a specific point in space, the end of the sweep route, without moving back into the sweep area to engage any surviving Red agents. The ANOVA for this measure is shown at Table 6.

**Table 6: ANOVA for MOE1: Time To Service Target Set (JMP Generated Output Table)**

**Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 7 | 1.37604e-6 | 1.9658e-7 | 186.6734 |
| Error | 352 | 3.70675e-7 | 1.0531e-9 | **Prob > F** |
| C. Total | 359 | 1.74671e-6 | | <.0001 * |

**Effect Tests**

| Source | Nparm | DF | Sum of Squares | F Ratio | Prob > F |
|---|---|---|---|---|---|
| NumMRM(0,4) | 1 | 1 | 9.7407e-10 | 0.9250 | 0.3368 |
| NumSRM(0,4) | 1 | 1 | 1.51327e-8 | 14.3703 | 0.0002 * |
| NumCUDAS(0,8) | 1 | 1 | 1.08828e-8 | 10.3345 | 0.0014 * |
| NumRedFighters(4,8) | 1 | 1 | 1.5821e-10 | 0.1502 | 0.6985 |
| BlueTactics | 1 | 1 | 1.09907e-6 | 1043.698 | <.0001 * |
| NumSRM*NumSRM | 1 | 1 | 5.02142e-9 | 4.7684 | 0.0296 * |
| NumCUDAS*NumCUDAS | 1 | 1 | 6.57629e-9 | 6.2450 | 0.0129 * |

For this set of data, at the 0.05 confidence level, the number of MRM and the number of Red Fighters do not have a statistically significant effect on the total time spent in the sweep area conducting the mission. However, all the other factors do significantly affect the mission time. Figure 36 shows the prediction profiler tool that JMP has. This allows us to quickly explore various scenarios of the factors, get the associated mean response predicted by the model and a prediction confidence interval

(also referred to as a prediction interval) on the results. For this measure, the original response data showed a non-constant variance problem in the residuals, which required an inverse transform to provide a more accurate model based on the data. The number reported in Figure 36 is read as the time in seconds because JMP automatically converts back from the inverse given by the model.

From Figure 36, we see that, in general, using more CUDA or more MRM missiles results in short mission times, whereas using more SRM tends to lengthen the mission times. Note that the effects of each factor on the mission time are very small compared to the choice of tactic. Use of the straight-in tactic is associated with significantly shorter mission times. This is primarily because the pincer tactic requires more time to set up a feasible shot at a Red Fighter.

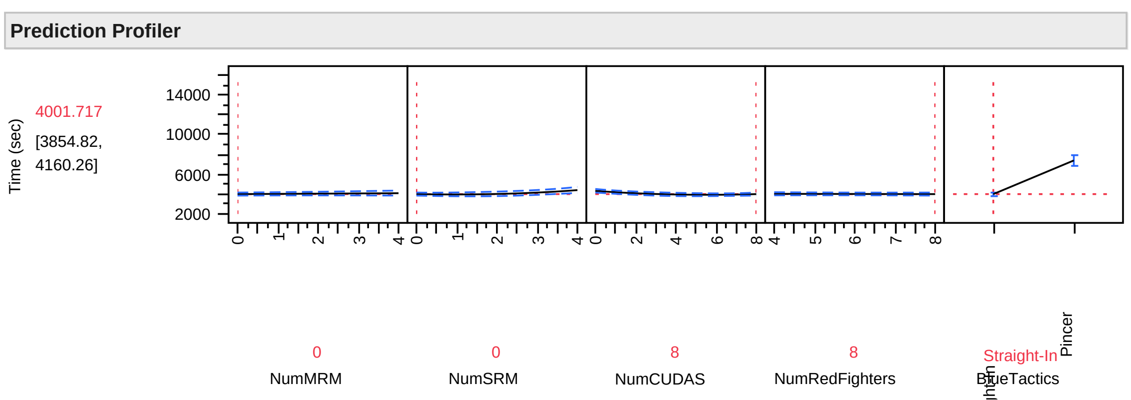



**Figure 36: Time as a function of each factor in the model (JMP Prediction Profiler)**

Tables 7 and 8 show times and 95% prediction intervals for various mixes of missiles with Blue using the straight-in tactic and pincer tactic, respectively. Both sets of data are for eight Red fighters in the mission area.

**Table 7: Mission Times for Blue Straight-In Tactic with 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Average Time (minutes) | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 73.72 | 70.78 | 76.92 |
| Mix 1 | 2 | 0 | 4 | 66.98 | 64.55 | 69.61 |
| Mix 2 | 0 | 2 | 4 | 66.60 | 64.25 | 69.13 |
| SACM Pure | 0 | 0 | 8 | 66.92 | 64.69 | 69.29 |
| MRM Pure | 4 | 0 | 0 | 74.19 | 71.44 | 77.17 |

**Table 8: Mission Times for Blue Pincer Tactic with 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Average Time | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 149.60 | 138.23 | 163.00 |
| Mix 1 | 2 | 0 | 4 | 124.24 | 116.30 | 133.34 |
| Mix 2 | 0 | 2 | 4 | 122.92 | 114.95 | 132.08 |
| SACM Pure | 0 | 0 | 8 | 124.00 | 115.85 | 133.38 |
| MRM Pure | 4 | 0 | 0 | 151.55 | 141.44 | 163.22 |

Note that these figures are from the constructed model used in the experimental design analysis. These averages should not be taken as actual performance. The most important use for these numbers is to show the differences between the mixes of weapons systems so that meaningful insights can be drawn about how the systems compare to each other. Figure 37 depicts the information shown in Tables 7 and 8.

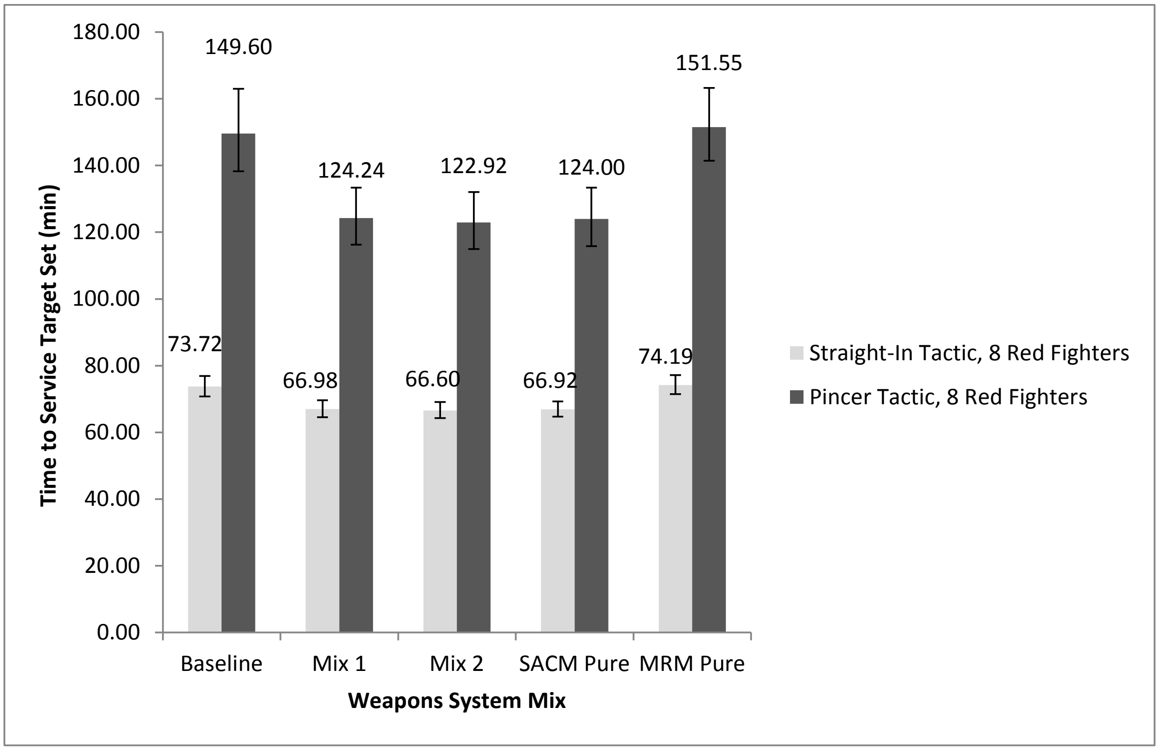



**Figure 37: Time to Service Target Set/Reach final sweep route destination for various mixes of weapons**

Figure 37 illustrates that use of the pincer tactic incurs a large increase in the average mission time over the use of straight-in tactics. Despite the larger prediction intervals for the pincer tactic, none of the intervals overlaps, meaning that the difference between the two tactics is statistically significant at the 0.05 confidence level. The large increase for the pincer tactic is also a practical difference, almost doubling the average mission time of using the straight-in tactic.

Any mix of weapons that includes the CUDA missile (Mix 1, Mix 2, and SACM pure) does effect a statistically significant decrease in mission time, but a very small decrease in comparison with the large difference due to choice in tactics. It can be argued whether the shorter mission times due to the CUDA have any practical difference.

**MOE2: Percent Targets Destroyed.**

The ANOVA for percent of targets destroyed is shown in Table 9. All of the factors appear to have some level of significance in determining the proportion of targets destroyed, although the choice of tactic is borderline significant, as depicted in Figure 39 to have a very small effect on the proportion of Red targets destroyed by Blue. A second order term in the number of CUDA is also showing significance.

**Table 9: ANOVA Percent of Targets Destroyed**

**Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 6 | 11.946159 | 1.99103 | 74.9618 |
| Error | 353 | 9.375875 | 0.02656 | Prob > F |
| C. Total | 359 | 21.322034 | | <.0001 ˆ |

**Effect Tests**

| Source | Nparm | DF | Sum of Squares | F Ratio | Prob > F |
|---|---|---|---|---|---|
| NumMRM(0,4) | 1 | 1 | 1.4772606 | 55.6186 | <.0001 ˆ |
| NumSRM(0,4) | 1 | 1 | 2.0059906 | 75.5252 | <.0001 * |
| NumCUDAS(0,8) | 1 | 1 | 6.4945023 | 244.5168 | <.0001 * |
| NumRedFighters(4,8) | 1 | 1 | 2.6378017 | 99.3128 | <.0001 * |
| BlueTactics | 1 | 1 | 0.1026186 | 3.8636 | 0.0501 |
| NumCUDAS*NumCUDAS | 1 | 1 | 0.4076256 | 15.3470 | 0.0001 * |

Figure 38 shows that generally increasing the number of each missile increases the relative percentage of Red destroyed. The straight-in tactic seems to indicate greater percentage of targets destroyed than use of the pincer tactic. Interestingly, as the number of Red fighters increase the proportion of targets that Blue is able to destroy decreases.

Additionally, there appears to be some curvature in the effect due to number of CUDA. This may indicate a diminishing return in that there is less increase in percentage of destroyed targets with increasing CUDA.

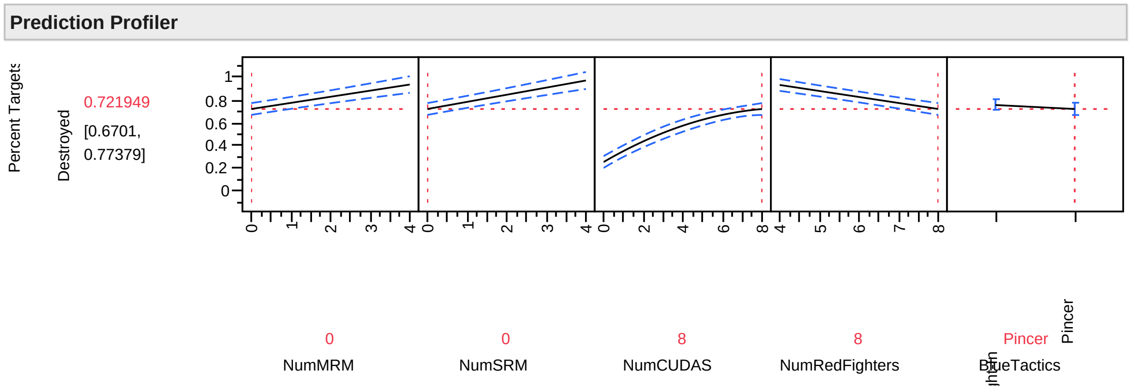



**Figure 38: Proportion of Targets Destroyed as a function of each factor (JMP Prediction Profiler)**

Tables 10 and 11 show the proportions of targets destroyed for various mixes of weapons for the straight-in tactic and pincer tactic respectively and number of Red fighters held constant at eight. This model has some borderline issues with non-constant variance within the residuals that may mean other methods of analysis are needed for this proportional data, including consideration of higher order ANOVA models and experimental designs with more design points and more replications to control variance. Again, these numbers are not meant as indicative of actual performance in a sweep scenario; rather, they are more useful in comparison to each other.

**Table 10: Percent of Targets Destroyed; Blue Straight-In Tactic with 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Proportion of Targets Destroyed | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 0.52 | 0.48 | 0.56 |
| Mix 1 | 2 | 0 | 4 | 0.72 | 0.67 | 0.76 |
| Mix 2 | 0 | 2 | 4 | 0.74 | 0.69 | 0.78 |
| SACM Pure | 0 | 0 | 8 | 0.76 | 0.71 | 0.80 |
| MRM Pure | 4 | 0 | 0 | 0.51 | 0.46 | 0.55 |

**Table 11: Percent of Targets Destroyed; Blue Pincer Tactic with 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Proportion of Targets Destroyed | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 0.49 | 0.45 | 0.52 |
| Mix 1 | 2 | 0 | 4 | 0.68 | 0.64 | 0.73 |
| Mix 2 | 0 | 2 | 4 | 0.70 | 0.65 | 0.75 |
| SACM Pure | 0 | 0 | 8 | 0.72 | 0.67 | 0.77 |
| MRM Pure | 4 | 0 | 0 | 0.47 | 0.43 | 0.51 |

Figure 39 depicts the information in Tables 10 and 11. Undoubtedly, mixes that include the CUDA perform significantly better over the MRM mixes, improving the proportion of targets destroyed by around 20%, which is both statistically significant (no overlap of the 95% prediction intervals) and realistically useful. However, there does not appear to be any statistically significant difference between each of the CUDA mixes at the 0.05 confidence level. Intriguingly, this also applies to the choice of tactic. Neither the pincer nor the straight-in tactic produces a significant difference in the proportion of the target set destroyed.

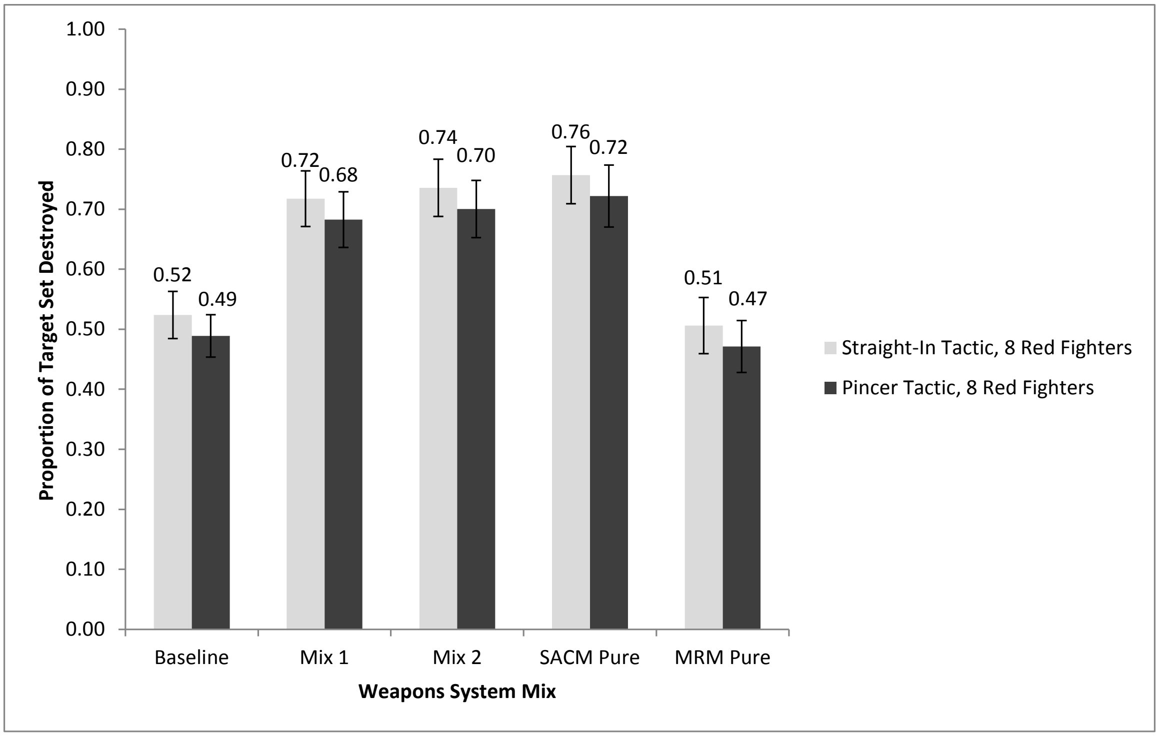



**Figure 39: Proportion of Red Targets Destroyed by Blue for various weapon mixes**

**MOE 3A Weapon Effectiveness.**

MOE 3A is calculated by dividing the total number of weapons fired by the total number targets destroyed. This calculation has an advantage of capturing the effectiveness of ground weapons due to the interaction of air-to-air combat outcomes. We further conducted analysis on the total number of air to air weapons fired by Blue divided the total number of air targets destroyed in order to provide information specific to the air combat portion of the scenario. We call this Air Weapon Effectiveness (MOE3B) in the next section.

The ANOVA again required a transformation on the response for the total weapon effectiveness number. The ANOVA, Table 12, is for a reduced model with an inverse transform on the response. Note that two second-order terms are included here to show

that there appears to be curvature in the weapon effectiveness response due to the CUDA and SRM.

**Table 12: ANOVA for Total Weapon Effectiveness Response**

**Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 7 | 6.623674 | 0.946239 | 61.3429 |
| Error | 350 | 5.398888 | 0.015425 | Prob > F |
| C. Total | 357 | 12.022562 | | <.0001 * |

**Effect Tests**

| Source | Nparm | DF | Sum of Squares | F Ratio | Prob > F |
|---|---|---|---|---|---|
| NumMRM(0,4) | 1 | 1 | 4.4322529 | 287.3348 | <.0001 * |
| NumSRM(0,4) | 1 | 1 | 3.5793994 | 232.0459 | <.0001 * |
| NumCUDAS(0,8) | 1 | 1 | 0.6809022 | 44.1416 | <.0001 * |
| NumRedFighters(4,8) | 1 | 1 | 0.2748019 | 17.8149 | <.0001 * |
| BlueTactics | 1 | 1 | 1.2082887 | 78.3311 | <.0001 * |
| NumSRM*NumSRM | 1 | 1 | 0.0963315 | 6.2450 | 0.0129 * |
| NumCUDAS*NumCUDAS | 1 | 1 | 0.1942966 | 12.5959 | 0.0004 * |

Figure 40 shows the weapon effectiveness contributions by each factor. Generally, with increasing number of missiles, there is an increase in the number weapons fired per target.  The straight-in tactic seems to show an increase of weapons used per target over the pincer tactic.  Although the number of Red fighters is statistically significant, it does not have a large effect on the weapon effectiveness.

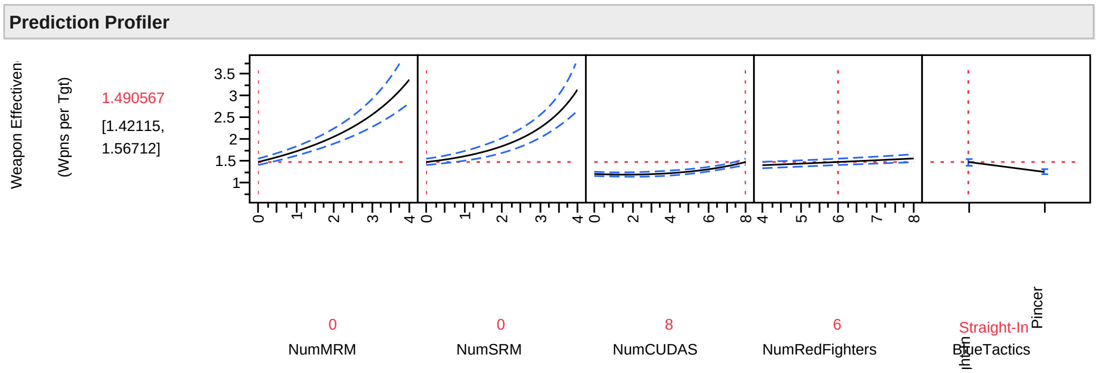



**Figure 40: Total Weapon Effectiveness as a function of each factor (JMP Prediction Profiler)**

Table 13 and Table 14 show average weapon effectiveness numbers for the various mixes using the straight-in tactic and the pincer tactic, respectively. Of note, the pincer lowers the number of weapons used per target significantly, particularly when CUDA is used. CUDA, and any mix involving CUDA, lowers number of weapons used per target significantly in both cases. Figure 41 illustrates these differences more clearly. What this means from a practical standpoint, because it is hard to envision decimals of a missile, is that on average, fewer missiles are used to produce the same amount of damage. It is certainly questionable whether the decrease is militarily useful, but we argue later that the decrease contributes to a decrease in risk and average mission times.

**Table 13: Total Weapon Effectiveness; Blue Straight-In Tactic, 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Total Wpn Effectiveness (Wpns Fired/Tgt Dest) | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 2.13 | 1.96 | 2.34 |
| Mix 1 | 2 | 0 | 4 | 1.70 | 1.60 | 1.82 |
| Mix 2 | 0 | 2 | 4 | 1.55 | 1.46 | 1.65 |
| SACM Pure | 0 | 0 | 8 | 1.57 | 1.49 | 1.67 |
| MRM Pure | 4 | 0 | 0 | 2.43 | 2.23 | 2.66 |

**Table 14: Total Weapon Effectiveness; Blue Pincer Tactic, 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Total Wpn Effectiveness (Wpns Fired/Tgt Dest) | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 1.70 | 1.59 | 1.82 |
| Mix 1 | 2 | 0 | 4 | 1.41 | 1.34 | 1.49 |
| Mix 2 | 0 | 2 | 4 | 1.31 | 1.24 | 1.38 |
| SACM Pure | 0 | 0 | 8 | 1.32 | 1.26 | 1.39 |
| MRM Pure | 4 | 0 | 0 | 1.88 | 1.77 | 2.00 |

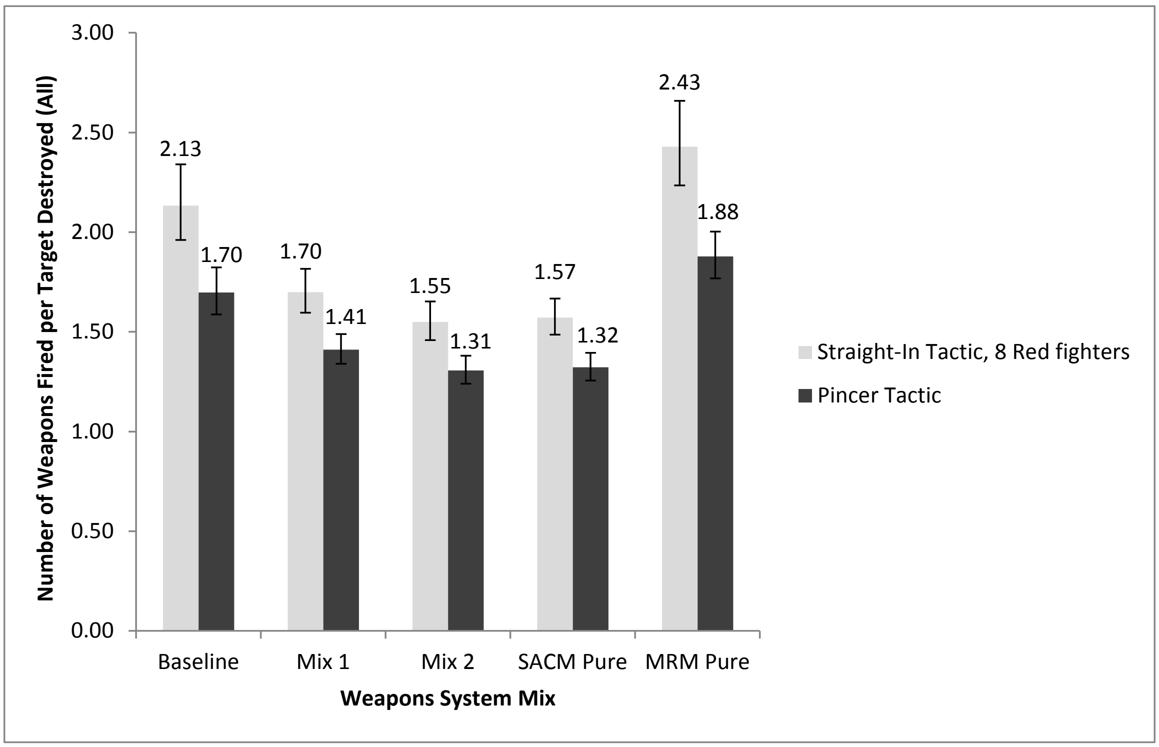



**Figure 41: Total Weapon Effectiveness for various mixes of weapons**

**MOE 3: Air-to-Air Weapon Effectiveness.**

The ANOVA model for this MOE had problems with normality in the residuals and non-constant variance of the residuals. To fix this, a transform is applied to the

response: $\ln\,(response + 1)$.  The number of Red fighters is not a significant factor and is removed from the model.

**Table 15: ANOVA Air Weapon Effectiveness**

**Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 5 | 96.45348 | 19.2907 | 236.4445 |
| Error | 348 | 28.39213 | 0.0816 | **Prob > F** |
| C. Total | 353 | 124.84561 | | <.0001 * |

**Effect Tests**

| Source | Nparm | DF | Sum of Squares | F Ratio | Prob > F |
|---|---|---|---|---|---|
| NumMRM(0,4) | 1 | 1 | 80.305995 | 984.3040 | <.0001 * |
| NumSRM(0,4) | 1 | 1 | 65.886593 | 807.5666 | <.0001 * |
| NumCUDAS(0,8) | 1 | 1 | 26.579873 | 325.7873 | <.0001 * |
| BlueTactics | 1 | 1 | 2.663681 | 32.6485 | <.0001 * |
| NumCUDAS*NumCUDAS | 1 | 1 | 2.614612 | 32.0471 | <.0001 * |

Tables 16 and 17 show average air-to-air weapon effectiveness for various mixes using the straight-in tactic and pincer tactic, respectively.  The pincer tactic decreases the weapons per target, as does any mix including the CUDA.

**Table 16: Air-to-Air Weapon Effectiveness; Straight-In Tactic**

| Mix Name | Number MRM | Number SRM | Number CUDA | A2A Wpn Effectiveness (Wpns Fired/Tgt Dest) | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 3.88 | 3.60 | 4.17 |
| Mix 1 | 2 | 0 | 4 | 2.06 | 1.84 | 2.29 |
| Mix 2 | 0 | 2 | 4 | 1.83 | 1.63 | 2.05 |
| SACM Pure | 0 | 0 | 8 | 1.75 | 1.55 | 1.96 |
| MRM Pure | 4 | 0 | 0 | 4.27 | 3.88 | 4.68 |

**Table 17: Air-to-Air Weapon Effectiveness; Pincer Tactic**

| Mix Name | Number MRM | Number SRM | Number CUDA | A2A Wpn Effectiveness (Wpns Fired/Tgt Dest) | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 3.08 | 2.88 | 3.29 |
| Mix 1 | 2 | 0 | 4 | 1.56 | 1.37 | 1.75 |
| Mix 2 | 0 | 2 | 4 | 1.37 | 1.20 | 1.55 |
| SACM Pure | 0 | 0 | 8 | 1.30 | 1.11 | 1.50 |
| MRM Pure | 4 | 0 | 0 | 3.41 | 3.10 | 3.73 |

Figure 42 shows that, although the pincer tactic does decrease the weapons used per air target, including the CUDA in the weapon mix decreases the weapons per target significantly for both tactics. Again, the combination of pincer and CUDA weapon mix uses the least number of weapons per target.

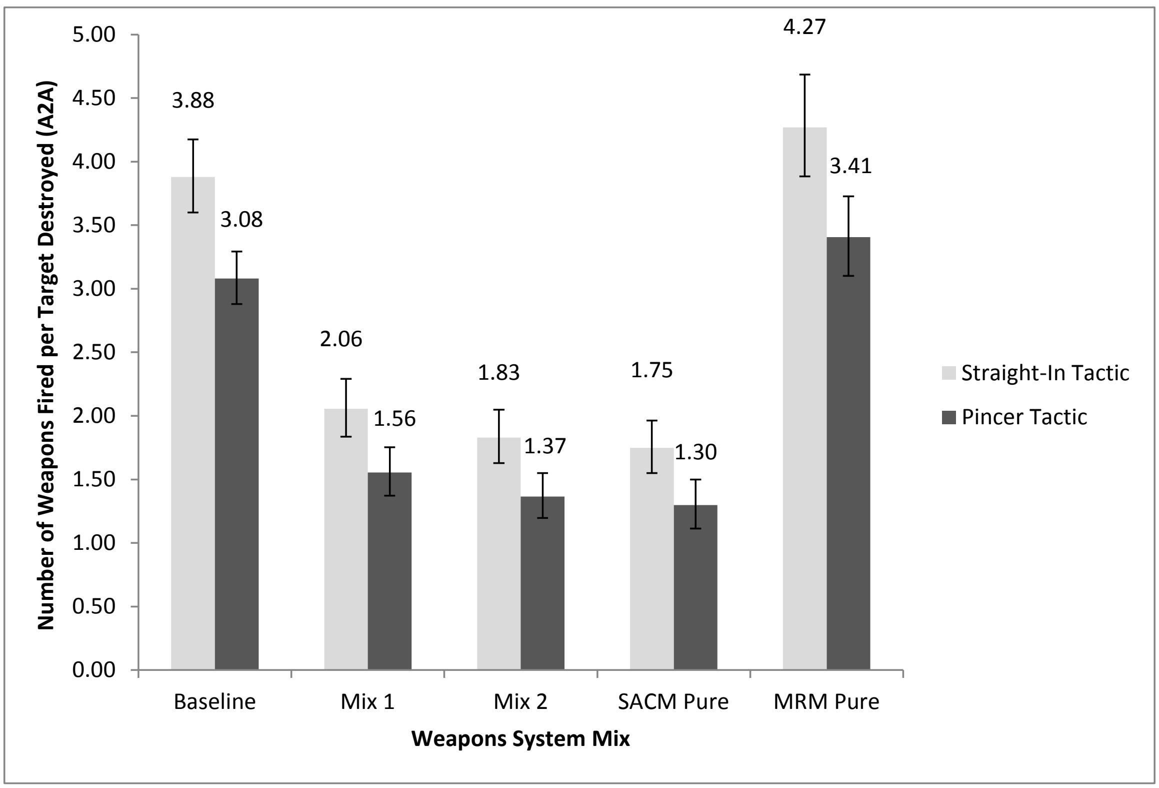



**Figure 42: Air to Air Weapon Effectiveness for various mixes of weapons**

**MOE 4: Average Engagement Distance**

The average engagement distance gives an idea of where engagements are generally taking place and how safe the Blue fighters are relative to the targets. Red weapons become more effective closer in; therefore, standoff generally is viewed as decreasing risk associated with air combat.

The ANOVA model for this MOE requires a square root transformation on the response to fix normality and non-constant variance issues on the residuals. The ANOVA is shown in Table 18. The number of MRM is not significant in the model, but is kept, as it is one of the main factors we are investigating. Additionally, the number of Red fighters is significant statistically, but does not provide a large effect on the average engagement distance.

**Table 18: ANOVA Average Engagement Distance**

**Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 7 | 315017.34 | 45002.5 | 166.3999 |
| Error | 350 | 94656.73 | 270.4 | **Prob > F** |
| C. Total | 357 | 409674.07 | | <.0001 * |

**Effect Tests**

| Source | Nparm | DF | Sum of Squares | F Ratio | Prob > F |
|---|---|---|---|---|---|
| NumMRM(0,4) | 1 | 1 | 386.939 | 1.4307 | 0.2325 |
| NumSRM(0,4) | 1 | 1 | 79842.795 | 295.2244 | <.0001 * |
| NumCUDAS(0,8) | 1 | 1 | 15461.669 | 57.1706 | <.0001 * |
| NumRedFighters(4,8) | 1 | 1 | 2427.759 | 8.9768 | 0.0029 * |
| BlueTactics | 1 | 1 | 51437.431 | 190.1936 | <.0001 * |
| NumSRM*NumSRM | 1 | 1 | 4509.499 | 16.6742 | <.0001 * |
| NumCUDAS*NumCUDAS | 1 | 1 | 5947.050 | 21.9896 | <.0001 * |

Table 19 and Table 20 show various engagement distances, as calculated by the ANOVA model, with 95% prediction intervals for the straight-in tactic and the pincer tactic, respectively, with number of Red fighters held constant at eight. CUDA seems to

significantly increase average engagement distances when it is included in the mix. The SACM pure (CUDA) mix and Mix 1, involving the CUDA and MRM, are significantly greater than all the mixes. A visualization of these numbers is depicted in Figure 43.

**Table 19: Average Engagement Distances; Straight-In Tactic with 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Average Engagement Distance (km) | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 25.56 | 23.83 | 27.34 |
| Mix 1 | 2 | 0 | 4 | 39.57 | 37.59 | 41.59 |
| Mix 2 | 0 | 2 | 4 | 33.94 | 32.00 | 35.94 |
| SACM Pure | 0 | 0 | 8 | 40.30 | 38.38 | 42.26 |
| MRM Pure | 4 | 0 | 0 | 30.46 | 28.84 | 32.14 |

**Table 20: Average Engagement Distances; Pincer Tactic with 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Average Engagement Distance (km) | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 18.23 | 16.80 | 19.72 |
| Mix 1 | 2 | 0 | 4 | 30.30 | 28.61 | 32.05 |
| Mix 2 | 0 | 2 | 4 | 25.41 | 23.70 | 27.18 |
| SACM Pure | 0 | 0 | 8 | 30.94 | 29.12 | 32.82 |
| MRM Pure | 4 | 0 | 0 | 22.41 | 21.12 | 23.74 |

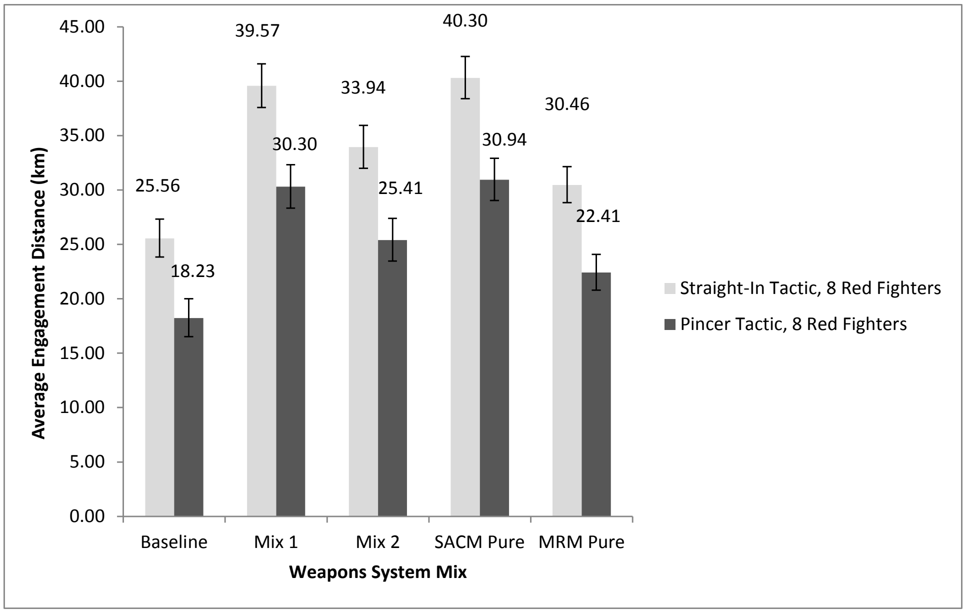



**Figure 43: Average Engagement Distance in kilometers for various mixes of weapons**

### MOE5: Number of Hits on Blue

The ANOVA for the number of hits on Blue is shown in Table 21. The response required a cube root transformation of the form $(response + 1)^{0.\overline{3}}$ due to normality and non-constant variance problems in the residuals. All the factors are significant, including second-order terms for number of Red Fighters and the number of CUDA. As a caution, the final model for this MOE has a low $R^2_{adj}$ number and significant lack of fit ($R^2_{adj} = 0.4759$). This means that there are most likely additional factors influencing the number of times that Blue aircraft are hit that are not considered in this study. For now, the model is sufficient to provide insight into how the factors interact with respect to the Blue agent's vulnerability within the mission area.

**Table 21: ANOVA Number of Hits on Blue**

**Analysis of Variance**

| Source | DF | Sum of Squares | Mean Square | F Ratio |
|---|---|---|---|---|
| Model | 7 | 36.375917 | 5.19656 | 47.5759 |
| Error | 352 | 38.447792 | 0.10923 | **Prob > F** |
| C. Total | 359 | 74.823709 | | <.0001 * |

**Effect Tests**

| Source | Nparm | DF | Sum of Squares | F Ratio | Prob > F |
|---|---|---|---|---|---|
| NumMRM(0,4) | 1 | 1 | 0.764739 | 7.0014 | 0.0085 * |
| NumSRM(0,4) | 1 | 1 | 4.547048 | 41.6295 | <.0001 * |
| NumCUDAS(0,8) | 1 | 1 | 9.587299 | 87.7743 | <.0001 * |
| NumRedFighters(4,8) | 1 | 1 | 15.551415 | 142.3774 | <.0001 * |
| BlueTactics | 1 | 1 | 2.943887 | 26.9521 | <.0001 * |
| NumCUDAS*NumCUDAS | 1 | 1 | 0.930008 | 8.5145 | 0.0037 * |
| NumRedFighters*NumRedFighters | 1 | 1 | 1.004532 | 9.1968 | 0.0026 * |

Tables 22 and 23 show the number of hits on Blue fighters for the straight-in tactic and the pincer tactic, respectively. Mixes that include CUDA show a significant decrease in how many hits Blue suffers over the other two cases. To reiterate, there are additional factors affecting the number of hits that Blue receives, such as passive threat detectors, pilot risk avoidance measures, etc. This model only considers specific factors associated with studying the effects of the mixes of offensive weapons arrayed on the Blue agents.

Generally, it can be seen in Figure 44 that the number of hits that Blue takes increase with use of the pincer over a straight-in tactic. The cases pictured in Figure 44 are those with the number of Red fighters held to eight. The model, intuitively, outputs less hits on Blue for less Red fighters, due to fewer chances for Red to fire at Blue (not depicted here for brevity).

**Table 22: Number of Hits on Blue; Straight-In Tactic with 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Number of Hits on Blue | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 8.87 | 7.58 | 10.29 |
| Mix 1 | 2 | 0 | 4 | 5.60 | 4.42 | 6.94 |
| Mix 2 | 0 | 2 | 4 | 4.31 | 3.28 | 5.50 |
| SACM Pure | 0 | 0 | 8 | 4.73 | 3.67 | 5.94 |
| MRM Pure | 4 | 0 | 0 | 10.83 | 9.12 | 12.74 |

**Table 23: Number of Hits on Blue; Pincer Tactic with 8 Red Fighters**

| Mix Name | Number MRM | Number SRM | Number CUDA | Number of Hits on Blue | Lower | Upper |
|---|---|---|---|---|---|---|
| Baseline | 2 | 2 | 0 | 12.32 | 10.87 | 13.90 |
| Mix 1 | 2 | 0 | 4 | 8.23 | 6.73 | 9.93 |
| Mix 2 | 0 | 2 | 4 | 6.59 | 5.25 | 8.12 |
| SACM Pure | 0 | 0 | 8 | 7.13 | 5.67 | 8.80 |
| MRM Pure | 4 | 0 | 0 | 14.73 | 12.77 | 16.88 |

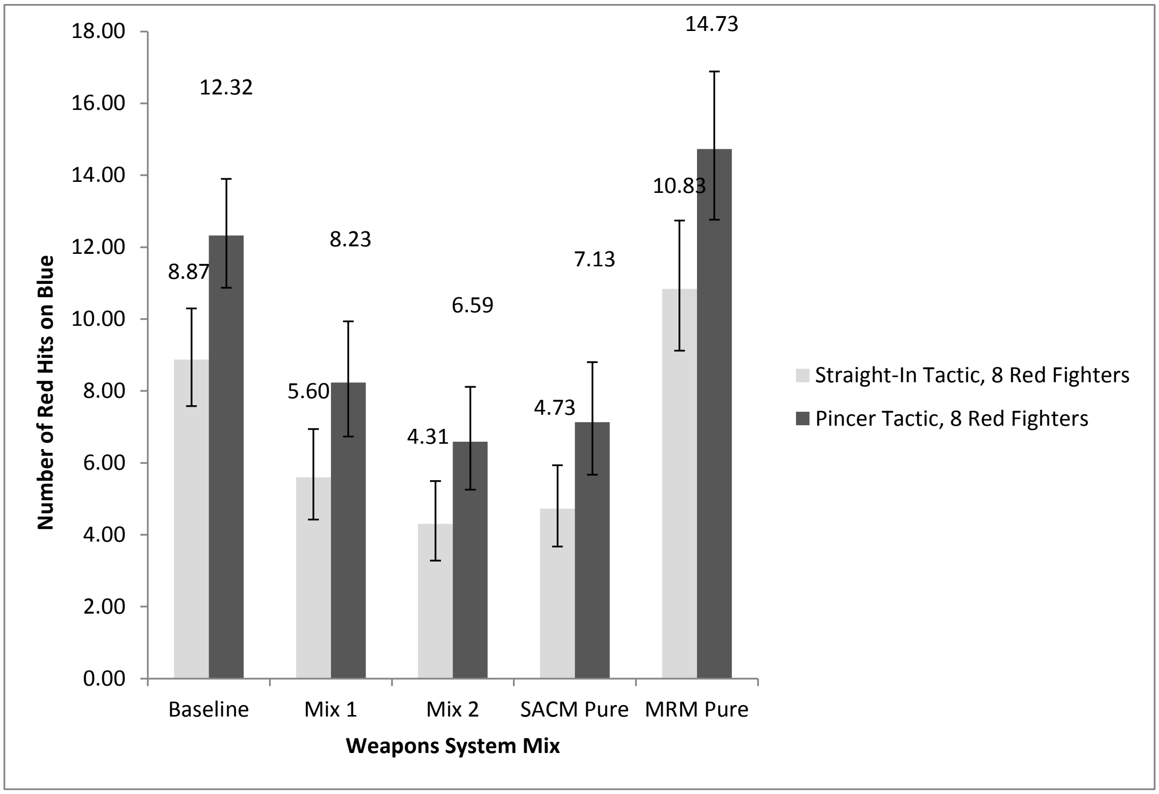



**Figure 44: Number of Red weapon hits on Blue agents for various Blue weapon mixes**

### Investigative Questions Answered

1. **What is the benefit of being able to carry more missiles? How does size/weight of missile affect mission outcomes?**

Weapon system load out mixes for fourth generation fighter aircraft that include the new missile technology (CUDA/SACM) significantly decreases times to service target set over the mixes without (MOE 1), but with a very small effect compared to the choice of tactic. Most of the decrease in average mission time is likely related to the higher effectiveness of the CUDA weapons mixes over the other mixes. Blue aircraft

using a CUDA mix need to use less weapons per target, meaning that there is less time spent re-engaging after an initial engagement was unsuccessful.

Mixes including the new missile technology show a significant increase in proportion of a target set destroyed for the sweep mission (MOE 2). This increase was large for CUDA mixes, meaning that when the new missile is used the overall mission effectiveness increases considerably. In mission scenarios, such as the sweep mission, that have follow on missions using the cleared mission corridor to execute a strategic target, this increase can have an impact.

Mixes including the CUDA show a decrease in the number of weapons used per target destroyed for both the total target set and for the air target set over mixes that do not include the CUDA (MOE 3A and 3B). In terms of mission effectiveness, a greater portion of the target set for the mission is destroyed more efficiently using mixes of weapons systems that include the CUDA. This means there are fewer weapons used to yield a greater number of destroyed targets and is due to the improved single shot kill probabilities, maneuverability, and range of the new missile. As mentioned, this contributes to faster mission completion. Higher effectiveness also helps decrease risk to the pilot and aircraft. The more maneuvering a fighter must do against a target, for instance if the first shot fails, the more likely the fighter will be shot at by the enemy aircraft. Admittedly, the decrease in weapons per target is small, but it is statistically significant. From a risk standpoint, any decrease, even a small decrease, in risk is desirable and can have practical significance depending on the situation.

Mixes including the CUDA show statistical evidence of decreasing the number of hits on Blue aircraft over mixes not including the CUDA (MOE 5). The significant

increase in standoff distance gained by use of the CUDA (MOE 4) contributes to this decrease in hits on Blue. Tactics and weapons that increase engagement distances help reduce risk to the aircraft able to employ these tactics and weapons.

Finally, the benefit of carrying a greater number of a smaller sized missile has diminishing returns in terms of weapon effectiveness and efficiency and is highly dependent on the number of enemy aircraft/ground targets within the mission area. Our sweep scenario is a simple one, but the number of enemy fighters in this scenario does not stress a fourth generation fighter with the ability to carry twice as many or three times as many of the new missile technology. In fact, we artificially lower the number of missiles available on the Blue aircraft in order to use the simulation to construct a suitable statistical model of effects of the missile system mixes. In other words, the Blue fighters should carry what is expected to be needed for a specific mission. If more missiles can be carried because of their lighter weight and smaller size, then it is also true that more cargo (fuel, electronic warfare equipment, etc.) can be carried because of the savings in weight if fewer missiles are carried. The main benefit in this type of new missile in terms of carrying capacity is in the flexibility it adds to the mission planning and load-out of a flight of fourth generation fighters.

2. **What is the proper mix of weapons? How does mission mode (air-to-air, air-to-ground) affect mission outcomes? Is there a benefit to carrying a mix of weapons?**

Clearly, MOE 1, 3A, 3B show significant improvement, statistically, for CUDA/SACM pure weapons mixes. Generally, carrying a mix of weapons shows no improvement over the pure weapon options. However, there are circumstances in which

short range air to air may be necessary, such as if, through maneuvering to gain advantage, the Blue finds itself at extremely close range, where use of CUDA or MRM may not be possible. Gun weapons systems were not modeled but could be an answer for extreme short range as well.

**3. What new tactics are possible given new weapon characteristics? Do tactics change over the range of each of the characteristics of the new missile type?**

Generally, pincer tactics results in longer mission times, smaller proportion of targets destroyed, shorter average engagement distances, and increased number of hits on Blue agents for all mixes (MOE 1, 2, 4, 5). Intuitively, mission times are lower for tactics like straight-in that move directly to contact as opposed to tactics like the pincer that take time for the aircraft to maneuver wide of the enemy aircraft into flanking positions. Use of the CUDA in conjunction with either of the tactics did not seem to produce particularly inflated effects on the average mission time. CUDA weapons mixes did decrease mission times somewhat more combined with the pincer tactic, but probably not enough to be useful when compared with the decrease in mission time observed with the straight-in tactic.

The pincer tactic did significantly decrease the number of weapons used per target destroyed for both all targets and air targets (MOE 3A, 3B). Combinations of CUDA weapons mixes and use of the pincer tactic actually provided the least number of weapons used per target.

The average engagement distance is significantly increased for CUDA/SACM mixes over other mixes (MOE 4), and as discussed this provides a benefit in terms of risk

reduction.  Tactics most suited to take advantage of this increased standoff capability are tactics such as the lead/trail tactic discussed in Chapter 3.

Of note, the extreme short fall in the pincer tactic is partially the result of two aspects of our simulation model.  One is that no AWACS is included in the scenario so that the Blue agents receive no early warning and no constant update of threat locations and actions.  The second is that the Blue aircraft are not outfitted with passive warning sensors to alert of incoming missiles.  Rather, the Blue agents in the simulation rely solely on their fire control radars to detect all objects in the air and a threat processor that "tells" the agent if any of the tracks sensed by the radar is an incoming missile.  Because of these two characteristics of the simulation scenario, the Blue agents lose situational awareness during a pincer as they turn to move to the flank of detected Red fighters.

Regardless of the characteristics of our scenario, the pincer tactic does take time to develop and is slower than moving straight to contact.  As discussed above, the CUDA missile has the ability to provide more flexible engagement options in terms of range and target aspect angles.  Tactics that attempt to maintain a BVR (Beyond Visual Recognition) engagement, such as the lead/trail tactic, can benefit from use of a CUDA weapon mix.  A combination of the lead/trail tactic and some type of flanking maneuver may even prove advantageous.  This combination could provide standoff while allowing the Blue side to use a portion of its force on the longer flanking move.

**Summary**

The new missile technology investigated in this simulation study shows some clear advantages.  Although the model of the CUDA used in this simulation is an

unclassified approximation of the true missile characteristics, the scenario showed that mixes using the CUDA improved in nearly every category over mixes including just medium range missiles or MRMs and short-range missiles. CUDA increased average engagement distances and decreased number of weapons used per target, which both contribute to a reduction in risk. CUDA mixes also exhibit a practical increase in proportion of the target set killed, useful if the area needs to be swept as clear as possible of designated enemy air and ground targets. CUDA mixes did statistically lower average mission times, however the realistic effect is very small compared with the effect due to choice of tactic and may not be useful for consideration as a benefit. Tactics best suited to the new missile are ones that maintain BVR to take advantage of the increased engagement ranges and possibly combined tactics that allow the flexible maneuvering characteristics of the new missiles to engage enemy aircraft at angles that the enemy aircraft will be unable to counter.

# V. Conclusions and Recommendations

## Review of the Weapon Systems Methodology Developed

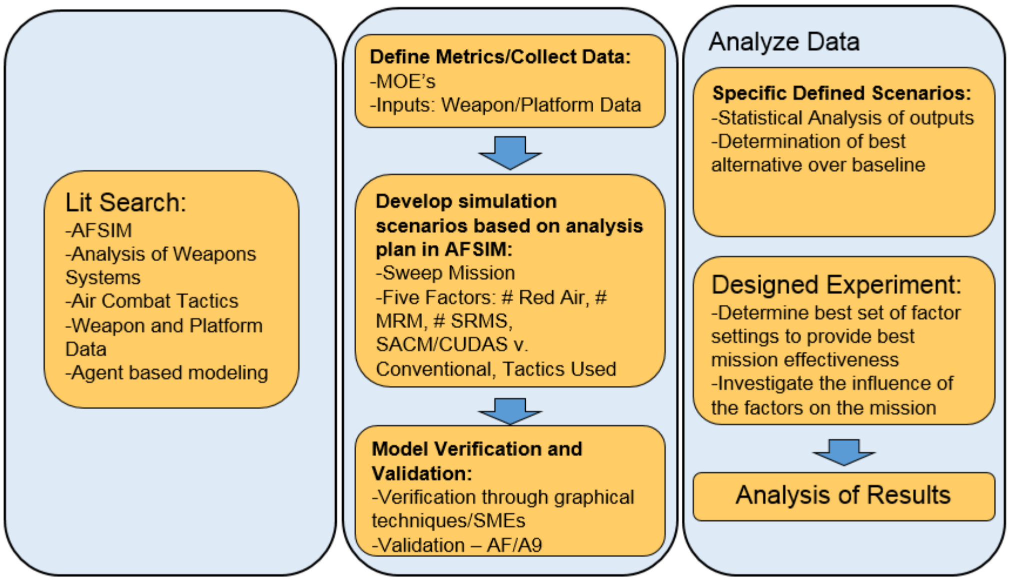



**Figure 45: Simulation Study Methodology for the Weapon System Analysis**

At its core, the methodology used in this analysis is a study using simulation as a designed experiment to investigate the benefits of a new weapon system. Figure 45 demonstrates the high-level study steps taken as a framework for execution of this analysis. The literature search involves researching published works and interviewing subject matter experts in order to determine the scope of the problem and formulate a problem statement. This leads into forming Measures of Effectiveness (MOEs) that allow us to distinguish the benefits of the new weapons system. For this study, the MOE's chosen are the average mission time, the proportion of the target set destroyed, a weapons effectiveness indicator, the average engagement distance, and the number of hits that the Red force is able to make against our Blue force.

Within the framework of the MOEs, a measurement space is developed that allows us to choose particular scenarios in which to test our new weapon system and set certain conditions that provide effects that the weapon system may interact with in order to supply us with information about the weapon system performance. Within this step, we also designate the factors we wish to investigate. These are the controlled, or decision, variables. In our study, the factors are the number of MRM, the number of SRM, the number of CUDA (the new weapon system of interest), the number of Red fighter aircraft, and the tactic used. The scenario we chose was the sweep scenario for several reasons. It fit the scope of demonstrating a simulation analysis methodology using a designed experiment. The scenario has all the aspects of air combat and many of the situations in which the new weapon system may be used. It is also scalable, from simple to complex, less opposing forces to more, etc. The sweep scenario is well defined in several sources, as discussed in Chapter 2. Finally, the Air Force Research Laboratory RQ division had recently developed the scenario within the AFSIM simulation framework. There are many scenarios, such as defensive counter air, airborne weapons layer, etc., that should also be used as a part of a comprehensive investigation of this missile system.

The next step in the study was to develop the scenario within a simulation environment. We chose AFSIM, for many reasons detailed in previous chapters, but particularly because of the simulation framework's object oriented nature. AFSIM has been in use for over ten years and has an extensive library of models that can be used in a simulation scenario, from aerodynamics and weapons effects models to pilot behavioral models. More importantly, scenarios, platforms, equipment, weapons and sensors

models previously developed for other scenarios and studies can quickly be adapted to the current study. For our purposes, the sweep scenario that AFRL/RQ developed needs little adaptation. We added more Red fighters, changed the behavioral engines of the various agents involved in the scenario, and added a few weapons models, specifically the CUDA, JDAM, and SRM models. However, the additional weapons systems needed very little work. We simply took previously defined missiles and adapted them to the specific characteristics of the new weapon system. This is the power of object oriented simulation models such as AFSIM.

Once the simulation model is constructed, it must be verified and validated. For the purposes of this study most of the verification is conducted through repeated visualization and adjustment of the scripts governing the simulation. Minimal validation was performed using subject matter experts to provide unofficial quality checks. Validation of many of the base models contained in the libraries in AFSIM is a much larger process and is currently being conducted by AF/A9.

After a valid model is ready, an experiment is designed using the factors defined in an earlier step. Because of the complex nature of the factors involved in this study, the JMP statistical package was used to provide a custom design as discussed in Chapter 4. We run the simulation model at each treatment combination, or design point, in the experimental design matrix as shown in Table 24. In order to provide a better statistical model, one that has a solid estimate of error, 20 replicates for each of the 18 design points are run. Each response recorded corresponds to one of the MOEs.

**Table 24: JMP Custom D-Optimal Design Matrix**

| Run | Number of MRM | Number of SRM | Number of CUDAS | Number of Red Fighters | Blue Tactic Used |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 4 | Pincer |
| 2 | 4 | 0 | 0 | 4 | Pincer |
| 3 | 2 | 2 | 0 | 4 | Pincer |
| 4 | 0 | 4 | 0 | 6 | Pincer |
| 5 | 2 | 0 | 4 | 6 | Pincer |
| 6 | 0 | 2 | 4 | 6 | Pincer |
| 7 | 4 | 0 | 0 | 8 | Pincer |
| 8 | 0 | 4 | 0 | 8 | Pincer |
| 9 | 0 | 0 | 8 | 8 | Pincer |
| 10 | 0 | 4 | 0 | 4 | Straight-In |
| 11 | 0 | 2 | 4 | 4 | Straight-In |
| 12 | 0 | 0 | 8 | 4 | Straight-In |
| 13 | 0 | 0 | 0 | 6 | Straight-In |
| 14 | 4 | 0 | 0 | 6 | Straight-In |
| 15 | 0 | 0 | 8 | 6 | Straight-In |
| 16 | 0 | 0 | 0 | 8 | Straight-In |
| 17 | 2 | 2 | 0 | 8 | Straight-In |
| 18 | 2 | 0 | 4 | 8 | Straight-In |

As soon as the simulation runs are made, the data is collected using some sort of post processor to translate the simulation output into usable numbers. For our study, we developed a post processor for AFSIM that uses the comma-delimited files output by AFSIM's simulation control engine. The post processor has a Microsoft Excel front end that uses Visual Basic for Applications (VBA) to call R scripts that parse the comma delimited text files, calculates the specific data required for the MOEs, and then places that data in a format more accessible to further statistical analysis. R has a very useful data structure, called a data frame, and many powerful functions that can slice and summarize the data in a data frame. R also has some very useful statistical packages, including design of experiments analysis, though none are used in this study.

The response data calculated from the post processor is then used to build statistical models using analysis of variance techniques (ANOVA) that the designed

experiment is specifically meant for. The designed experiment combined with the ANOVA provides the most amount of information for the least amount of design points and replications possible. For the statistical analysis of the experimental data, we employed JMP to build statistical models for each of the responses or MOEs. We have reported the results in Chapter 4 and briefly summarize them again here in section 5.2.

The final step is to report our insights. This paper is the culmination of the study and is the report that shows not only our analysis results, but also details the methodology used in order to further DOE and agent-based modeling approaches to analysis of new weapon systems.

**Summary of Findings and Insights**

The main benefit of following a designed experiment approach to analysis of an agent-based model of a complex system is that the resulting statistical models can be used to fully explore the factor space for each desired MOE. This exploration yields interesting insights that answer the main questions, but also provide potential avenues for further experimentation. For example, in our study, we discovered that the factors of number of Red fighters and number of missiles only interact with each other when the levels are scaled such that there are not an overwhelming number of missiles (Blue offensive capability) compared to the number of Red targets. In reality, a flight of fighters would always be sent out with capability to overmatch the enemy. However, this real world missile configuration does not give us very much information about how changes in the levels of numbers of missiles carried effects the outcomes of the mission.

For our study, further experimentation is needed to discover the effects due to a large increase in the number of CUDA by increasing the complexity of the mission scenario.

Another insight about the methodology used for studying this new weapon system is that the factor space involved in the air combat is complex. Experimental designs to explore this space using agent based modeling techniques must be carefully analyzed and compared before implementation.

Finally, the behavior engines used to drive agent behavior are very useful for building a complex environment full of agent interaction in order to closely approximate complex air combat systems. This allows us to capture information about comparative performance of the different weapons mixes that may not be present in a simpler simulation devoid of more complex agent decision-making behavior.

As for the analysis conducted on the new weapon system, our statistical models show that we can truly show the significance of different factors effects within air combat. For example, we are able to show a both a statistically significant difference and a militarily useful difference in the proportion of target set destroyed. As a contrast, our results also show that use of the CUDA is statistically significant in driving down average mission times, but the amount by which those times are decreased, on average, may not have a practical significance. Still, these types of conclusions allow us to glimpse more information about how the system works and find benefits that we may have only hypothesized.

## Recommendations for Future Research

### Use of Advanced Artificial Intelligence and Machine Learning Methods.

One avenue for future research we suggest is implementation of the Unified Behavior Framework (UBF), discussed in Chapter 2, within the complex scenario. The research would have the goal of discovering more useful tactics to employ with the new weapon system. By allowing agents in the simulation to make more complex decisions that have an element of learning, the emergent behaviors can be captured and analyzed to show the range of tactical options that may be paired with the new weapon system.

Additionally, different algorithms for the deliberative functions of the behavior framework should be tried. For instance, heuristic algorithms, such as tabu search or simulated annealing that provide near optimal rather than optimal solutions to the weapon-target assignment problem, may provide a closer approximation to how pilots make this critical decision in reality. Learning technologies may also be implemented in the deliberative layer to allow the agents to update their tactics according to the state of the environment and to learn what tactics work best. This method of agent behavior at the deliberative layer may provide more emergent behaviors to study for better tactics to use with the new weapon system.

### Analysis of the New Missile System in Alternative Scenarios

Our study focused on only one of the mission roles for which this new weapon system can possibly be used. Future research should include analysis of the weapon system in several different scenarios. We suggest, at a minimum, defensive counter-air and airborne weapons layer scenarios.

Additionally, more advanced fighters and different types of fighters and sensors should be included in future scenarios. For instance, an AWACS should be included in at least one scenario to provide more information on how situational awareness may affect the choice of tactic used.

**Weapon System Cost Benefits**

In today's budget constrained environment, costs are a very important component of the analysis of any weapon system. To provide a comprehensive analysis, future research must include cost analysis in terms of fuel costs, procurement, and life-cycle costs. For instance, smaller, lighter missiles may produce less fuel consumption over mission distances.

**Effects of Capability to Carry Large Number of New Missiles**

As discussed in Chapter 4, this study limited the ratio of missiles carried to number of Red air targets in order to bring the two numbers more into parity. To address this, we slightly increased the number of Red air targets and halved the number of missiles the Blue fighters carried. Further research should investigate more complex scenarios with very large numbers of Red fighters to show if the trends of increasing the new weapons system provides similar benefits over much different measurement space. For instance, there was a diminishing return on the proportion of the target set destroyed for an increase in the number of new missiles. The investigation may reveal that this holds over a much different target set or that there is a different relationship as the number of targets increases drastically.

## Conclusion

We have presented a methodology for conducting analysis of a new weapon system under consideration for different mission roles. The main elements of this methodology include a designed simulation experiment, agent-based modeling and artificial intelligence techniques, and basic data analysis techniques. The simulation provides insights in the stochastic nature of the complex systems under investigation. Building a simulation using semi-autonomous agents induces further complexity that more closely mirrors the complex system that combat represents. A designed experiment provides a wealth of information on the factors and response of the complex system to help us discover meaningful insights into the system and the benefits of using the new weapon system. Finally, statistical analysis shows the how the various components interact with each other and provides a method to compare different possibilities throughout the total space of factor combinations.

# Appendix A: Example AFSIM comma delimited output file

| | run | time | event | platform | target | weapon | engagement.number | start.time | shooter.Lat | shooter.Lon |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 741.88667 | WEAPON_FIRED | air_interceptor_1 | Blue_2 | air_intercepto | 1 | 741.88667 | 36.7711 | -122.477 |
| 2 | 1 | 771.02 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 2 | 771.02 | 37.2566 | -122.318 |
| 3 | 1 | 791.02 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 3 | 791.02 | 37.2566 | -122.318 |
| 4 | 1 | 816.02 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 4 | 816.02 | 37.2566 | -122.318 |
| 5 | 1 | 836.02 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 5 | 836.02 | 37.2566 | -122.318 |
| 6 | 1 | 861.02 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 6 | 861.02 | 37.2566 | -122.318 |
| 7 | 1 | 881.02 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 7 | 881.02 | 37.2566 | -122.318 |
| 8 | 1 | 906.02 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 8 | 906.02 | 37.2566 | -122.318 |
| 9 | 1 | 926.02 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 9 | 926.02 | 37.2566 | -122.318 |
| 10 | 1 | 1164.74333 | WEAPON_FIRED | air_interceptor_2 | Blue_1 | air_intercepto | 10 | 1164.74333 | 36.9455 | -121.899 |
| 11 | 1 | 1166.69333 | WEAPON_FIRED | air_interceptor_1 | Blue_1 | air_intercepto | 11 | 1166.69333 | 36.9253 | -121.882 |
| 12 | 1 | 1179.94333 | WEAPON_FIRED | air_interceptor_2 | Blue_1 | air_intercepto | 12 | 1179.94333 | 36.9686 | -121.915 |
| 13 | 1 | 1181.89333 | WEAPON_FIRED | air_interceptor_1 | Blue_1 | air_intercepto | 13 | 1181.89333 | 36.9485 | -121.898 |
| 14 | 1 | 1195.14333 | WEAPON_FIRED | air_interceptor_2 | Blue_1 | air_intercepto | 14 | 1195.14333 | 36.9909 | -121.933 |
| 15 | 1 | 1197.09333 | WEAPON_FIRED | air_interceptor_1 | Blue_1 | air_intercepto | 15 | 1197.09333 | 36.9713 | -121.917 |
| 16 | 1 | 1545.60333 | WEAPON_FIRED | air_interceptor_2 | Blue_1 | air_intercepto | 16 | 1545.60333 | 36.8071 | -122.284 |
| 17 | 1 | 1553.61333 | WEAPON_FIRED | air_interceptor_1 | Blue_1 | air_intercepto | 17 | 1553.61333 | 36.8003 | -122.275 |
| 18 | 1 | 1560.80333 | WEAPON_FIRED | air_interceptor_2 | Blue_1 | air_intercepto | 18 | 1560.80333 | 36.7887 | -122.257 |
| 19 | 1 | 1658.60333 | WEAPON_FIRED | air_interceptor_1 | Blue_1 | air_intercepto | 19 | 1658.60333 | 36.7156 | -122.148 |
| 20 | 1 | 1673.80333 | WEAPON_FIRED | air_interceptor_1 | Blue_1 | air_intercepto | 20 | 1673.80333 | 36.7435 | -122.142 |
| 21 | 1 | 1831.68667 | WEAPON_FIRED | long_sam_1 | Blue_1 | long_sam_1_s | 21 | 1831.68667 | 36.6 | -121 |
| 22 | 1 | 1851.68667 | WEAPON_FIRED | long_sam_1 | Blue_1 | long_sam_1_s | 22 | 1851.68667 | 36.6 | -121 |
| 23 | 1 | 1876.68667 | WEAPON_FIRED | long_sam_1 | Blue_1 | long_sam_1_s | 23 | 1876.68667 | 36.6 | -121 |
| 24 | 1 | 1896.68667 | WEAPON_FIRED | long_sam_1 | Blue_1 | long_sam_1_s | 24 | 1896.68667 | 36.6 | -121 |
| 25 | 1 | 2082.60333 | WEAPON_FIRED | air_interceptor_2 | Blue_1 | air_intercepto | 25 | 2082.60333 | 36.8095 | -122.28 |
| 26 | 1 | 2097.80333 | WEAPON_FIRED | air_interceptor_2 | Blue_1 | air_intercepto | 26 | 2097.80333 | 36.7902 | -122.254 |
| 27 | 1 | 2144.25333 | WEAPON_FIRED | air_interceptor_2 | Blue_1 | air_intercepto | 27 | 2144.25333 | 36.7154 | -122.197 |
| 28 | 1 | 2265.95667 | WEAPON_FIRED | air_interceptor_1 | Blue_2 | air_intercepto | 28 | 2265.95667 | 36.9148 | -121.886 |
| 29 | 1 | 3074 | WEAPON_FIRED | Blue_1 | aggregate_sh | Blue_1_jdam_ | 29 | 3074 | 37.271 | -121.534 |
| 30 | 1 | 3192.18667 | WEAPON_FIRED | long_sam_1 | Blue_1 | long_sam_1_s | 30 | 3192.18667 | 36.6 | -121 |
| 31 | 1 | 3212.18667 | WEAPON_FIRED | long_sam_1 | Blue_1 | long_sam_1_s | 31 | 3212.18667 | 36.6 | -121 |
| 32 | 1 | 3237.18667 | WEAPON_FIRED | long_sam_1 | Blue_1 | long_sam_1_s | 32 | 3237.18667 | 36.6 | -121 |
| 33 | 1 | 3257.18667 | WEAPON_FIRED | long_sam_1 | Blue_1 | long_sam_1_s | 33 | 3257.18667 | 36.6 | -121 |
| 34 | 1 | 5898 | WEAPON_FIRED | Blue_2 | air_intercepto | Blue_2_aim-9 | 34 | 5898 | 36.8898 | -121.916 |
| 35 | 1 | 5930 | WEAPON_FIRED | Blue_2 | air_intercepto | Blue_2_aim-9 | 35 | 5930 | 36.9397 | -121.902 |
| 36 | 1 | 6896 | WEAPON_FIRED | Blue_1 | long_sam_2 | Blue_1_jdam_ | 36 | 6896 | 37.0222 | -122.259 |
| 37 | 2 | 748.55333 | WEAPON_FIRED | air_interceptor_2 | Blue_2 | air_intercepto | 1 | 748.55333 | 36.7546 | -122.511 |
| 38 | 2 | 751.54 | WEAPON_FIRED | air_interceptor_1 | Blue_2 | air_intercepto | 2 | 751.54 | 36.7246 | -122.481 |
| 39 | 2 | 769.68667 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 3 | 769.68667 | 37.2566 | -122.318 |
| 40 | 2 | 778.92333 | WEAPON_FIRED | air_interceptor_2 | Blue_2 | air_intercepto | 4 | 778.92333 | 36.7053 | -122.542 |
| 41 | 2 | 779.9 | WEAPON_FIRED | air_interceptor_1 | Blue_2 | air_intercepto | 5 | 779.9 | 36.6788 | -122.514 |
| 42 | 2 | 789.68667 | WEAPON_FIRED | long_sam_2 | Blue_1 | long_sam_2_s | 6 | 789.68667 | 37.2566 | -122.318 |
| 43 | 2 | 807.28333 | WEAPON_FIRED | air_interceptor_2 | Blue_2 | air_intercepto | 7 | 807.28333 | 36.6555 | -122.563 |

**Figure 46: Weapon Event Data Output from AFSIM simulation run, first 11 columns out of 45 columns in original file**

# AGENT-BASED MODELING METHODOLOGY FOR ANALYZING WEAPON SYSTEMS

**MAJ Casey D. Connors**
**Advisor: Dr. J.O. Miller, PhD**
**Reader: LTC Brian Lunday, PhD**
Department of Operational Sciences (ENS)
Air Force Institute of Technology

## Introduction:

- Focus of Study: How new missile should be used in the air combat environment at the tactical level
- Require a mission level scenario in a dynamic, stochastic simulation model
- Air combat modeled as a complex adaptive system using agent-based modeling
- Constrained to single, simplistic instance of a sweep mission scenario

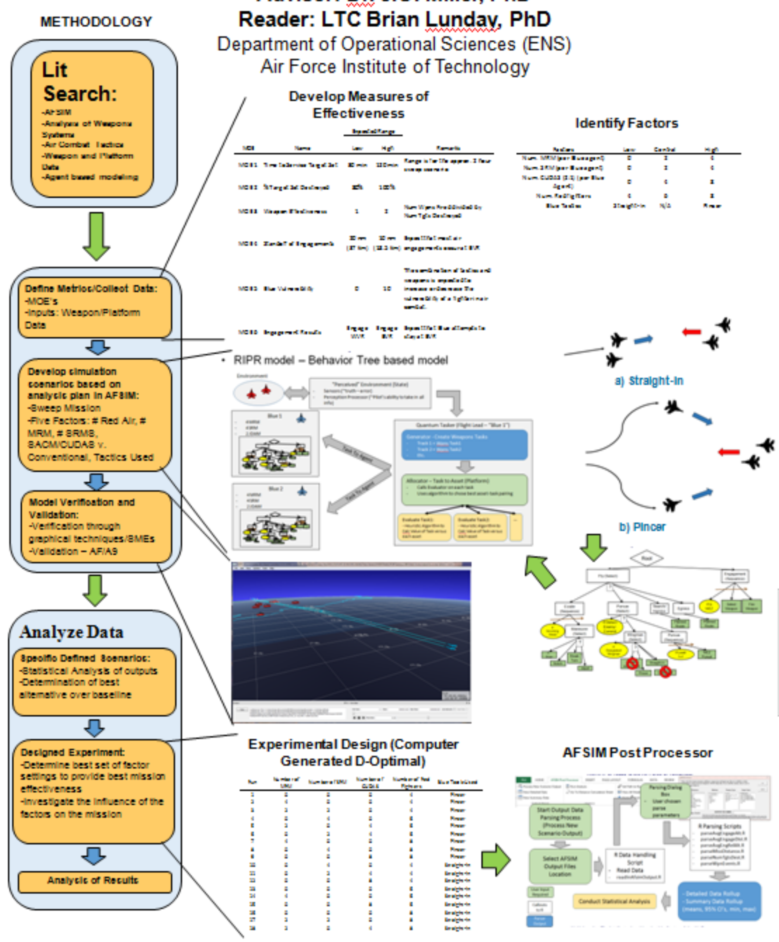
## Problem Statement:

- Develop an analysis methodology to determine effects of a new weapon on tactics and combat decision making by modeling flexible entity behaviors in simulation.

## Research Objectives:

- Develop methodology that applies to new platform delivered weapon systems and perhaps even new types of sensors and communications systems
- Methodology answers questions of benefits, appropriate weapon mixes, and range of tactics to use with weapon
- Demonstrate Analytic Framework for Simulation, Integration and Modeling (AFSIM)

AIR FORCE INSTITUTE of TECHNOLOGY

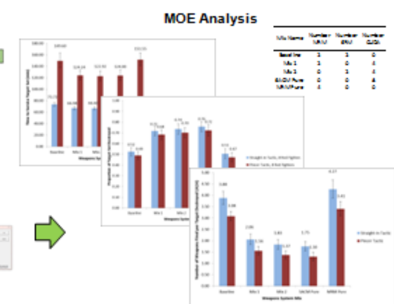



## Contributions:

- Presented methodology for analysis of the benefits of a new weapon system
- Agent-based modeling more closely models the complexity of modern air combat
- Designed experiment provides the most information about the air combat system in the fewest number of runs
- Many possible excursions can be analyzed based on one experiment
- Developed AFSIM Post Processor using R with an Excel GUI

## Future Research:

- Use of advanced artificial intelligence and machine learning methods
- Analysis of the new missile in alternative scenarios (including classified scenarios)/More complex scenarios
- Weapon system cost benefits
- Effects of capability to carry large number of missiles




Sponsor:
Lockheed Martin Missiles and Fire Control Systems
POC: Robert Cisneros

## Bibliography


Ahner, D., & Parson, C. (2013). Weapon Trade Off Analysis Using Dynamic Programming for a Dynamic Weapon Target Assignment Problem within a Simulation. *Proceedings of the 2013 Winter Simulation Conference (WSC).* Wright Patterson AFB OH.

Ahuja, R. K., Kumar, A., Jha, K. C., & Orlin, J. B. (2003). Exact and Heuristic Algorithms for the Weapon Target Assignment Problem. *Operations Research*.

Baker, J. R. (1986). Tactics and the Fog of War. *USAF Fighter Weapons Review*, 14-17.

Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2004). *Discrete-Event System Simulation.* Upper Saddle River, NJ: Pearson Prentice Hall.

Bechhofer, R. E., Elmaghraby, S., & Morse , N. (1958). A Single-Sample Multiple-Decision Procedure For Selecting the Mulitnomial Event Which Has the Highest Probability. *Sibley School of Mechanical Engineering, Cornell University*, 102-119.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology*. Cambridge MA: The MIT Press.

Bullock, R. K., McIntyre, G. A., & Hill, R. R. (2000). Using Agent-Based Modeling to Capture Airpower Strategic Effects. *Proceedings of the 2000 Winter Simulation Conference*, (pp. 1739-1746).

Chen, J., Xin, B., Peng, Z., Dou, L., & Zhang, J. (2009). Evolutionary decision-makings for the dynamic weapon-target assignment problem. *Science in China Series F: Information Sciences*, 2006-2018.

Corbett, M. (2013). A New Approach to Ballistic Missile Defense for Countering Antiaccess/Area-Denial Threats from Precision-Guided Weapons. *Air & Space Power Journal*, 83-106.

Cruz, J., Simaan, M., Gacic, A., Jiang, H., Letellier, B., Li, M., & Liu, Y. (2001). Game Theoretic modeling and control of a military air operation. *IEEE Transactions on Aerospace and Electronic Systems*, 1393-1405.

DOD Models and Simulation Coordination Office. (2014). *M&S Glossary*. Retrieved from DoD M&S Coordination Office: http://www.msco.mil/

Gat, E. (1998). On Three-Layer Architectures. *Artificial Intelligence and Mobile Robots*.

Geaslen, J., & Panson, D. (2014). Self Protection and Reactive Technology for an Advanced Combat Utility System (SPARTACUS) Air Combat Development Scenario. Wright Patterson Air Force Base OH: Air Force Research Laboratory Aerospace Technologies.

Hewson, R. (2009). *Jane's Air Launched Weapons Issue 53.* Alexandria VA: Jane's Information Group, Inc.

Hill, R. R., & McIntyre, G. A. (2001). Applications of Discrete Event Simulation Modeling to Military Problems. *Proceedings of the 2001 Winter Simulation Conference, 1*, 780-788.

Hill, R. R., McIntyre, G. A., & Narayanan, S. (2001). Genetic Algorithms for Model Optimization. *Proceedings of Simulation Technology and Training Conference (SimTechT).*

Houck, M. R., Whitaker, L. A., & Kendall, R. R. (1993). *An Information Processing Classification of Beyond-Visual-Range Air Intercepts.* Brooks Air Force Base, TX: Air Force Material Command.

Jackson, P., Munson, K., Peacock, L., Bushell, S., & Willis, D. (2013). *Jane's All the World's Aircraft.* Alexandria VA: Jane's Information Group, Inc.

Law, A. M. (2007). *Simulation and Modeling Analysis.* New York NY: McGraw-Hill.

Marzinotto, A., Colledanchise, M., Smith, C., & Ogren, P. (2014). Towards a Unified Behavior Trees Framework for Robot Control. *2014 IEEE International Conference on Robotics and Automation (ICRA).*

Middleton, V. (2010). Simulating Small Unit Military Operations with Agent-Based Models of Complex Adaptive Systems. *Proceedings of the 2010 Winter Simulation Conference, 1*, 119-134.

Miller, J. O., & Nelson, B. L. (1996). Getting More From the Data in Multinomial Selection Problem. *Proceedings of the 1996 Winter Simulation Conference.* Orlando FL.

Montgomery, D. C. (2009). *Design and Analysis of Experiments.* Hoboken, NJ: John Wiley & Sons, Inc.

Mulgund, S. S., Harper, K. A., Guarino, S. L., & Zacharias, G. L. (1999). *Situation Awareness Model for Pilot-in-the-Loop Evaluation.* Cambridge MA: Charles River Analytics.

Myers, R. H., Montgomery, D. C., & Anderson-Cook, C. M. (2009). *Response Surface Methodology.* Hoboken NJ: John Wiley & Sons, Inc.

Nuzzo, R. (2014). Statistical Errors: P-values, the 'gold standard' of statistical validity, are not as reliable as many scientists assume. *Nature*, *506*, 150-152.

Parunak, H. V., Savit, R., & Riolo, R. L. (1998). Agent-based modeling vs. equation-based modeling: A case study and users' guide. In *Multi-agent systems and agent-based simulation* (pp. 10-25). Heidelberg: Springer.

R Core Team. (2014). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org

Rood, J., Chilton, J., Campbell, K., & Jenkins, J. (2013). Missile Defense Perspectives Panel. *Space and Missile Defense Symposium.*

SAS Institute Inc. (1989-2007). JMP, Version 10.0. Cary NC.

Spronck, P., Ponsen, M., Sprinkhuizen-Kuyper, I., & Postma, E. (2006). Adaptive Game AI with Dynamic Scripting. *Machine Learning, 3*(63), 217-248.

Wild, C., & Seber, G. (1999). *Chance Encounters: A First Course in Data Analysis and Inference; Chapt. 10: Wilcoxon.* New York: John Wiley & Sons.

Woolley, B. G., & Peterson, G. L. (2009). Unified Behavior Framework for Reactive Robot Control. *Journal of Intelligent and Robotic Systems*, *55*(2), 155-176.

Zeh, J., & Birkmire, B. (2014). *Advanced Framework for Simulation, Integration and Modeling (AFSIM) Version 1.8 Overview.* Wright Patterson Air Force Base OH: Air Force Research Laboratory, Aerospace Systems.

| REPORT DOCUMENTATION PAGE | | | *Form Approved* *OMB No. 0704–0188* |
|---|---|---|---|
| The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS. | | | |
| 1. REPORT DATE (DD–MM–YYYY) 26-03-2015 | 2. REPORT TYPE Master's Thesis | | 3. DATES COVERED (*From — To*) Sept 2013 – Mar 2015 |
| 4. TITLE AND SUBTITLE Agent-based Modeling Methodology For Analyzing Weapons Systems | | | 5a. CONTRACT NUMBER<br>5b. GRANT NUMBER<br>5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) Connors, Casey D., Major, USA | | | 5d. PROJECT NUMBER ENY15734B<br>5e. TASK NUMBER<br>5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>Air Force Institute of Technology<br>Graduate School of Engineering and Management (AFIT/EN)<br>2950 Hobson Way<br>WPAFB OH 45433-7765 | | | 8. PERFORMING ORGANIZATION REPORT NUMBER<br>AFIT-ENS-MS-15-M-132 |
| 9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br>Robert Cisneros<br>Lockheed Martin Missiles and Fire Control<br>1902 West Freeway, Grand Prairie, Texas 75051<br>Robert.cisneros@lmco.com, 972-603-7352 | | | 10. SPONSOR/MONITOR'S ACRONYM(S) LMMFC<br>11. SPONSOR/MONITOR'S REPORT NUMBER(S) |
| 12. DISTRIBUTION / AVAILABILITY STATEMENT<br>Distribution Statement A. Approved for public release; Distribution Unlimited. | | | |
| 13. SUPPLEMENTARY NOTES<br>This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States. | | | |

14. ABSTRACT

Getting as much information as possible to make decisions about acquisition of new weapons systems, through analysis of the weapons systems' benefits and costs, yields better decisions. This study has twin goals. The first is to demonstrate a sound methodology to yield the most information about benefits of a particular weapon system. Second, to provide some baseline analysis of the benefits of a new type of missile, the Small Advanced Capability Missile (SACM) concept, in an unclassified general sense that will help improve further, more detailed, classified investigations into the benefits of this missile. In a simplified, unclassified scenario, we show that the SACM provides several advantages and we demonstrate a basis for further investigation into which tactics should be used in conjunction with the SACM. Furthermore, we discuss how each of the chosen factors influence the air combat scenario. Ultimately, we establish the usefulness of a designed experimental approach to analysis of agent-based simulation models and how agent-based models yield a great amount of information about the complex interactions of different actors on the battlefield.

15. SUBJECT TERMS
Agent-based modeling, simulation, combat modeling, weapons systems analysis, design of experiments

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON Miller, J.O., Dr. ADVISOR |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | 19b. TELEPHONE NUMBER (Include Area Code) (937) 255-6565, ext. 4326 John.miller@afit.edu |
| U | U | U | UU | 131 | |

**Standard Form 298 (Rev. 8–98)**
***Prescribed by ANSI Std. Z39.18***